LIFENET: A FLEXIBLE AD HOC NETWORKING SOLUTION FOR TRANSIENT ENVIRONMENTS

A Thesis
Presented to
The Academic Faculty

by

Hrushikesh S Mehendale
hrushi@gatech.edu

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Computer Science

Georgia Institute of Technology
December 2011

LIFENET: A FLEXIBLE AD HOC NETWORKING SOLUTION FOR TRANSIENT ENVIRONMENTS

Approved by:

Dr. Santosh Vempala,
Committee Chair
School of Computer Science
Georgia Institute of Technology

Dr. Santosh Vempala, Advisor
School of Computer Science
Georgia Institute of Technology

Dr. Ashok Jhunjhunwala
Professor, Electrical Engineering Dept.
IIT Madras, India

Dr. Michael Best
School of Computer Science
Georgia Institute of Technology

Dr. Nick Feamster
School of Computer Science
Georgia Institute of Technology

Date Approved: 3 August 2011

To ...

Aai, Baba, Aaji, Tatya, Rujuta and Sneha

ACKNOWLEDGEMENTS

Working with Prof. Santosh Vempala will remain one of my most cherished experiences. I can't thank him enough for presenting me an opportunity, the time, the freedom and all the necessary support, typically required for working on a difficult unsolved problem. In spite of being very busy, he always seemed to have enough time to discuss new ideas, debate new approaches and discuss related problems. Although he is a theoretical computer scientist by expertise, for LifeNet and rightly so, his focus has always been on implementation and evaluation. His insights have imparted a strong theoretical foundation to the otherwise experimental evolution of LifeNet.

Prof. Ashok Jhunjhunwala has always been an inspiration to me since I first worked with him before I came to Georgia Tech. I am completely convinced by his approach of taking research beyond the confines of a research paper and applying it on the field until it becomes self-sustainable. His comments and feedback were invaluable to the development of LifeNet and always steered us in the right direction. Now that LifeNet would be formally incorporated as a company, the support given by him makes us ever more confident.

I would also like to thank Prof. Nick Feamster and Prof. Mike Best for their time and being on the committee. Being a networking expert, Nick has always given frank comments and pointed out areas where we need to improve. Prof. Mike's rich experience in ICT4D makes him an invaluable resource person for us and would need more and more of his help as LifeNet gets deployed on field.

Ashwin Paranjpe has always been a very strong support even after he left school. He was the person who built the first prototype of LifeNet during the time he worked with Santosh. In fact, it was he who thought of 'LifeNet' as a name for our solution.

Although he stopped working on LifeNet long time ago, he is always available for discussing ideas and possible implementations. His well-thought-out approach and sound network-programming skills make him an ideal reviewer for any piece of network or systems code.

I am also grateful to Ranjita Bhagwan, my mentor at Microsoft Research. I had a great time working with her on an interesting research problem that combined machine learning with network troubleshooting. I particularly appreciate her inputs and various approaches that she suggested for presenting LifeNet as a research paper.

I would also like to thank Prof. Constantine Dovrolis and Prof. Ada Gavrilovska for their feedback on our work. I would also like to thank Prof. Mostafa Ammar for teaching the Graduate Level Computer Networking so exceptionally well. Attending his course lectures used to be a fascinating learning exercise. I particularly remember the seminal TCP modeling paper by Padhye et. al., that he made so easy for each of us to understand.

I am also thankful to my lab-mates Stephen Tyler, Supraja Narsimhan and Ruban Monu for their continuous feedback and ideas for the furtherment of LifeNet during the weekly C4G meetings. I also thank Yogesh Mundada particularly for reviewing many of our paper drafts. I would specially like to thank Elizabeth Ndongi and Dani Denton, without whom, working in Klaus would have been very difficult.

Thanks is definitely not what friends expect, but this section would not complete without the mention of Hrishikesh Pathak, Shruti Bhide, Mandar Gosavi, Kedar Chitale, Janhavi Modak, Alok Sabade, Chaitrali Amrutkar, Nikhil Mehta, Tushar Mahule and Karan Deshmukh for being there whenever needed.

Finally, this work would not have been possible without the love and support of my father , mother, sister, grand-parents and Sneha. They have believed in me and supported me all throughout my life and I am what I am only because of them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In the wake of major disasters, the failure of existing communications infrastructure and the subsequent lack of an effective communication solution results in increased risks, inefficiencies, damage and casualties. Currently available options such as satellite communication are expensive and have limited functionality. A robust communication solution should be affordable, easy to deploy, require little infrastructure, consume little power and facilitate Internet access. Researchers have long proposed the use of ad hoc wireless networks for such scenarios. However such networks have so far failed to create any impact, primarily because they are unable to handle network transience and have usability constraints such as static topologies and dependence on specific platforms.

LifeNet is a WiFi-based ad hoc data communication solution designed for use in highly transient environments. After presenting the motivation, design principles and key insights from prior literature, the dissertation introduces a new routing metric called Reachability and a new routing protocol based on it, called Flexible Routing. Roughly speaking, reachability measures the end-to-end multi-path probability that a packet transmitted by a source reaches its final destination. Using experimental results, it is shown that even with high transience, the reachability metric - (1) accurately captures the effects of transience (2) provides a compact and eventually consistent global network view at individual nodes, (3) is easy to calculate and maintain and (4) captures availability. Flexible Routing trades throughput for availability and fault-tolerance and ensures successful packet delivery under varying degrees of transience.

With the intent of deploying LifeNet on field we have been continuously interacting with field partners, one of which is Tata Institute of Social Sciences India. We have refined LifeNet iteratively refined base on their feedback. I conclude the thesis with lessons learned from our field trips so far and deployment plans for the near future.

CHAPTER I

INTRODUCTION

Wireless communication has now become an integral part of everybody's daily lives. This is quite evident from the fact that the number of mobile phone subsriptions worldwide has reached 4.6 billion [40], which is around 60 percent of the world's population today. Significant increase in bandwidth and recent advances in portable computing have together taken computing to a higher orbit.

1.1 Context

Although the evolution of wireless communication technologies has been quick, their overall architectures have not changed much. Cellular networks and WiFi networks are two of the most widely used wireless networks today and can be considered as representative examples. Architectures of both these networks are hierarchical with a clear top-down functional distribution.

In cellular networks, the network is divided into a number of cells (Figure 1). Each cell has its own Base Transceiver Station (BTS), which directly talks to all user-end mobile phones in its cell. A Base Station Controller (BSC) is responsible for a set of cells and talks to their respective $BTSs$. A Mobile Switching Center (MSC) typically lies at the root of the hierarchy and manages a set of $BSCs$. Technologies have evolved from GSM [32], CDMA [43] to 3G [44] and 4G [10], but the hierarchical architecture still persists.

WiFi is another type of network, which we use everyday. It usually consists of end-user devices such as laptops, smart-phones, etc. associated to a WiFi router. The WiFi router lies at the root of the hierarchy and manages the entire WiFi communication amongst the devices associated with it and with the outside Internet.

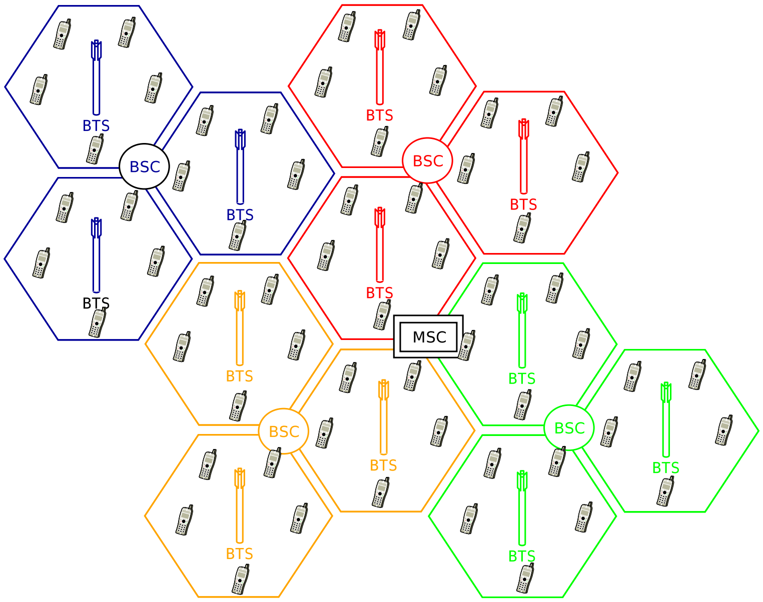


Figure 1: Architecture of a typical cellular network

These network architectures have some key disadvantages by design. The designs have been engineered for performance and efficiency; reliability and fault-tolerance have been given secondary importance. Due to their adherance to strict functional hierarchy, they seem to have evolved into single-point failure systems. Reliability is typically traded for *'Performance at Optimal Cost'*. For example in a cellular network, if MSC fails, communication is hampered in the entire network under it. If a BSC fails, the entire network that it manages, which may include a few $BTSs$ and several mobile phone users, would fail. If a BTS fails, all mobile phone users in its cell would not be able to communicate (unless there is another overlapping cell). Similarly in a WiFi network, if the WiFi router fails, end user-devices associated with it such as laptops and smart-phones would fail to communicate.

Secondly, these architectures are infrastructure-based. Each hierarchical level consists of one important node, which is solely responsible for managing the communication in the network under itself. BTS from a cellular network can be considered as an example. A BTS needs to talk to every mobile phone in its cell. The average radius of a cell is approximately 1 Km. In order to service the users in its cell 24X7, BTS needs infrastructure in the form of (a) a large power supply (and backup) and (b) high gain antennas mounted on towers (for achieving the required coverage). Due to heavy reliance on infrastructure, such an architecture cannot be used in areas where infrastructure does not exist or is partially or completely destroyed.

1.2 Motivation and the Problem

In spite of considerable advances in mobile wireless communication, one finds many scenarios in real life, that lack a dependable and affordable communication solution. Communication in the aftermath of disasters, communication in remote resource-constrained areas, communication in oil and natural gas exploration sites, on-ship maritime communication, on-field communication for media personnel, communication during trekking, mountaineering and archeological expeditions, wireless sensor networks, etc. are some representative examples of such real life scenarios.

There are two common constraints of all these afore-mentioned scenarios - (a) transience and (b) lack of infrastructure. By transience, I refer to the changing conditions along various dimensions such as node failures, mobile nodes, changing physical obstructions and interference. Since none of the existing communication technologies is designed to be infrastructure-free and reliable under transience, these scenarios are still deprived of affordable and reliable connectivity.

Researchers have long argued that ad hoc wireless networks are an ideal solution for such scenarios. First few ideas for ad hoc routing were proposed in mid to late 1990's ([38, 26, 37]). Today, even after a decade and a half of research and hundreds of publications, the problem of efficient routing in transient environments still remains unsolved. The first phase of research consisted of several new routing protocol proposals with some simulation based performance results. These early protocols were mostly variants of standard distance-vector ([1]) or link-state ([2]) routing protocols used in wired networks. In the second phase, researchers tried to implement these early protocols and published their experiences in the process. Out of the many insightful findings, researchers could surmise two key understandings - (1) Simulation results are far from the actual reality and proposed early protocols have many limitations and (2) Traditional routing metrics for e.g. hop-count are not suitable for wireless networks due to the inherent non-determinism in the wireless channel. In the third and current phase, researchers now equipped with deeper understandings of the problem, focused on implementations. New metrics ([12]) were proposed and new opportunistic routing protocols ([7, 13]) were implemented and evaluated. It is during this time that the capacity limitations on multihop ad hoc routing were understood. Even though the new routing approaches achieved substantial throughput improvements than early routing protocols, they had two key limitations due which we do not see them widely deployed in real life - (1) the throughput improvements were not good enough for high-bandwidth applications and (2) their designs had constrains such as static topology and lack of fault-tolerance, which made them unusable even for low-bandwidth applications under transience. For detailed literature review, please read Chapter 2.

1.3 LifeNet: A Solution

We argue that if the constraint of 'high-throughput' is relaxed, it is possible to realize ad hoc networks that are flexible and reliable under transience. For the scenarios mentioned above, easy and rapid establishment of baseline connectivity in highly transient environments is priority as against high throughput. For example, consider communication in the aftermath of disasters. Medium or large scale disasters usually hamper the communication infrastructure either by direct physical destruction or indirectly due to power failure. In such a situation, a network which can be easily setup with minimum infrastructure, which is reliable against power failures (failing nodes) and which handles moving nodes and changing physical obstructions, is very much the need of the day. The work presented on LifeNet in this dissertation demonstrates that it is possible to realize such infrastructure-free and fault tolerant networks at the expense of throughput.

Broadly speaking, this work has *two novel contributions*. The first contribution is a new routing metric called *'Reachability'*. Reachability accurately captures the effect of transience (mobile nodes, failing nodes, changing physical obstructions), is easy to compute and maintain, and enables a compact representation of the entire network at individual nodes, which facilitates routing. The second contribution is a new routing protocol based on the reachability metric, called *'Flexible Routing'*. Flexible routing is a multipath routing protocol that uses pairwise reachabilities to reliably deliver packets under varying degrees of transience. It trades throughput to achieve the required reliability in communication. This work borrows many concepts from the early work done by Ashwin and Santosh [34]. They proposed a framework called MyMANET for implementing mobile ad hoc networks. It used Virtual Distance as a routing metric, which was based on end-to-end packet loss. A primitive routing protocol based on Virtual Distance was also implemented. Reachability and the flexible routing protocol emerged and were concretized after extending and re-engineering MyMANET many times.

Three *design principles* (see Chapter 3) that guided us since the early days, proved to be the key factor behind the successful realization of our ideas. The first design principle was *'use of commodity hardware and systems'*. We always believed that the key for greater acceptance would be to build a solution which is interoperable with different hardware platforms and operating systems. Moreover, it is only by following this principle that we were able to achieve our goal of infrastructure-free

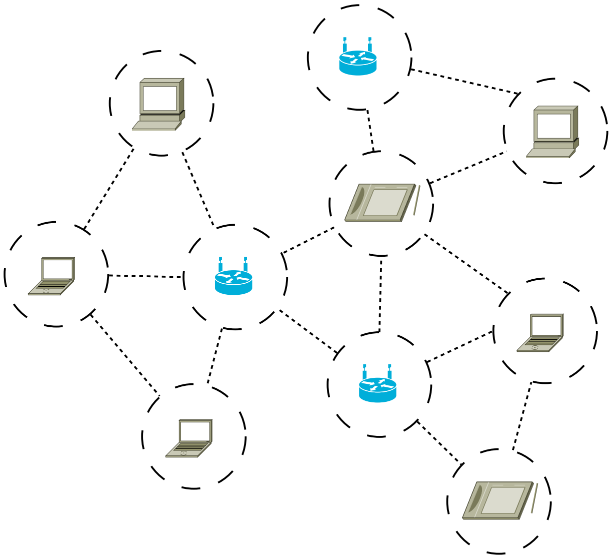


Figure 2: Proposed solution schematic

connectivity. The second principle was *'throughput can be traded for reliability and usability'*. This principle allowed flexible routing to be truly completely distributed and fault-tolerant by design. By trading efficiency (throughput) we were able to achieve the required reliability (fault-tolerance) for handling transience. The motivation behind this design principle is that since the capacity of multihop wireless networks is inherently insufficient, there is no harm in trading throughput for higher levels of reliability, flexibility and usability, if doing so promises to serve some critical needs. The third design principle was *'availability under eventual consistency'*. Maintaining consistency in topology information becomes extremely difficult as the network scales, particularly in transient conditions. We argue that in order to reliably route packets under varying degrees of transience, the routing protocol should not require strictly consistent topology information. The reachability metric enables a compact and easily maintenable representation (eventual consistent) of the entire network graph at individual nodes. The flexible routing protocol uses pairwise reachabilities to route packets using multiple available paths, successfully handling transience.

Reachability (defined in Chapter 4) is a directional metric, and captures the effect of transience in a single numerical value. Roughly speaking, it measures the end-to-end multipath probability that a packet transmitted by a source node reaches the destination node. In other words, reachability aims to measure the maximum number of ways by which a packet transmitted at the source node can reach the destination node within a fixed number of hops. Evaluation results show that this number is an accurate characterization of the network state as it is influenced by factors like changing topology, physical obstructions, traffic, interference, etc.

Flexible routing is designed as a pro-active routing protocol. Prior literature presents enough evidence that reactive approaches to routing do not work well in transient networks. The protocol uses pairwise reachabilities to route packets along multiple paths towards the destination. Multipath routing is essential for handling transience since it provides backup paths that can effectively handle node failures, topology changes caused by moving nodes and can route around interference or congestion hotspots. Needless to mention, multipath routing comes at the cost of reduction in throughput. Another important design decision is to not maintain routes or paths explicitly as it does not scale well. The core routing decision for flexible routing is *"Whether or not to forward?"* instead of *"Which node to forward the packet to?"*.

A rigorous evaluation of LifeNet was conducted in a university building environment. The evaluation of reachability and flexible routing was conducted with the intent of finding answers to these questions - (1) How accurately does reachability capture transience? and (2) How accurately and efficiently does the proposed flexible routing protocol utilize reachability to reliably deliver packets under transience?

We validated the following hypotheses during evaluation.

- Reachability efficiently captures the effect of mobility
- Reachability captures the phenomenon that connectivity of the network as a whole increases as the network scales

- Reachability captures the effect of degraded connectivity as node failures happen
- Flexible routing utilizes reachability to strengthen routing as the network scales
- Flexible routing utilizes reachability to gracefully degrade its performance as node failures happen
- Flexible routing maintains its performance in conditions of node mobility.

Please refer to Chapter 6) for more details on the evaluation of LifeNet.

The design of LifeNet has been an interative process. Our focus has always been on building a system that satisfies critical needs as against trying to fit an already existing solution into some existing problem. Users were hence involved from the early stages of system design. Chapter 7 details our efforts in forming partnerships for on-field deployment of LifeNet and lessons learnt until now from the initial field visits and surveys.

CHAPTER II

THE STORY OF AD HOC WIRELESS NETWORKING

The story of ad hoc wireless networking is no less exciting than a Tom and Jerry cat and mouse chase. Researchers being the big cats trying to tame ad hoc routing, which like Jerry seems very hard to catch! First few ideas for ad hoc routing were proposed in mid to late 1990's ([38, 26, 37]). Today, even after a decade and a half of research and hundreds of publications, the problem of efficient routing in transient environments still remains unsolved. The first phase of research consisted of several new routing protocol proposals with some simulation based performance results. These early protocols were mostly variants of standard distance-vector ([1]) or link-state ([2]) routing protocols used in wired networks. In the second phase, researchers tried to implement these early protocols and published their experiences in the process. Out of the many insightful findings, researchers could surmise two key understandings - (1) Simulation results are far from the actual reality and proposed early protocols have many limitations and (2) Traditional routing metrics for e.g. hop-count are not suitable for wireless networks due to the inherent non-determinism in the wireless channel. In the third and current phase, researchers now equipped with deeper understandings of the problem, focused on implementations. New metrics ([12]) were proposed and new opportunistic routing protocols ([7, 13]) were implemented and evaluated. It is during this time that the capacity limitations on multihop ad hoc routing were understood. Even though the new routing approaches achieved substantial throughput improvements than early routing protocols, they had two key limitations due which we do not see them widely deployed in real life - (1) the throughput improvements were not good enough for high-bandwidth applications and (2) their designs had constrains such as static topology and lack of fault-tolerance, which made them unusable even for low-bandwidth applications under transience.

Before delving deep into the literature review, clarification of some terms used throughout this dissertation is necessary.

2.1 Types of routing protocols

2.1.1 Distance Vector routing

Distance Vector routing protocols are the protocols in which, every individual node maintains ¡distance, vector¿ tuple information for all other nodes on the network. Distance of any node is the cost of reaching that node and vector is the name of the network interface to which, packets destined to that node, should be forwarded. Once a node joins an already existing network, it identifies its neighbours, listens for their routing updates and then populates its ¡distance, vector¿ table. Once on the network, individual nodes maintain topology information (i.e. distance table) by helping their respective neighbours by periodically providing distance updates. RIP [36] and IGRP [22] are two of the most widely used distance vector routing protocols that have been around since a long time.

2.1.2 Link State routing

Link State routing protocols are the protocols in which, every individual node needs to maintain a complete or a partial map of the entire network including the nodes and the connecting links. When a network link changes its state (ON or OFF), a notification, called a link state advertisement is flooded throughout the network. All other nodes on the network note the change and recompute their routes accordingly. This method is more reliable, easier to debug and less bandwidth intensive than distance vector routing. However it also more complex, and more CPU and memory intensive as well. OSPF [33] is an example of a widely used link state routing protocol.

2.2 Approaches to ad hoc routing

Primarily there are the following two approaches to routing in ad hoc networks - proactive routing and reactive routing. In ad hoc networks, nodes do not start with a prior knowledge of the network topology, instead, they have to discover it. The basic idea is that a new node may announce its presence and should listen for announcements broadcast by its neighbours. Each node learns about nodes nearby, and how to reach them, and may annouce that it, too, can reach them.

2.2.1 Proactive routing

In proactive routing, individual nodes maintain fresh list of distance vectors or routers by periodically distributing routing updates to a part or the entire network. In other words, topology information is precomputed before the actual data transfer may occur. DSDV [38] is an example of a proactive routing protocol. The key challenge in implementing proactive routing protocols under transience is to achieve a practical tradeoff between the conflicting goals of efficiently broadcasting routing updates and accurately maintaining topology information.

2.2.2 Reactive routing

In reactive routing, a source node finds and decides a route to the destination node on demand by flooding the network with Route Request packets. DSR [26] and AODV [37] are examples of reactive routing protocols. The key challenge in implementing reactive routing protocol is to quickly decide paths on demand and maintain them as nodes fail or move around in the network under transience.

As mentioned earlier research on wireless ad hoc networking can be chronologically grouped in phases.

2.3 Early protocols

DSDV [38] was the first formally proposed protocol for MANETs by Perkins and Bhagwat. It is a proactive routing protocol, which used hop count as the routing metric. For the first time it proposed the idea of modeling computers and end user devices as routers. The DSDV proposal was followed by the DSR [26] proposal. DSR, unlike DSDV was designed as a reactive protocol that calculate paths on demand. In DSR, routes are established on-demand just prior to data transmission and then serviced by route maintenance algorithms. Another protocol called AODV [37], improved upon DSR, retaining its flavour. ZRP [19] was designed as a hybrid protocol (both proactive and reactive) for large sized networks of nodes with varying degree of mobility. But some of its functionality was still based on DSDV. Another protocol called TORA [35] attempted to suppress the generation of far-reaching control messages by maintaining a directed-acyclic-graph rooted at the destination. It gave more emphasis on avoiding route-discovery and route optimality was secondary in importance. OLSR [4] used link state routing which required the maintenance of connectivity graph at every node. In a network with a dynamic topology, maintaining consistency of connectivity graph across the entire network and storing it at every node was an expensive proposal. In [9, 25, 11], performance of DSR, DSDV, AODV and TORA was evaluated using simulations. DSR and AODV generally performed better. However, performance results and comparisons of these early routing protocols were simulation based and were not repeated when they were implemented in practice. Work done later [12] showed that hop count is not a suitable metric for ad hoc networking as most links have intermediate loss rates. In [46], researchers showed that DSDV and AODV both could not provide stable paths when implemented.

2.4 Modeling and evaluation insights

Authors of [18] showed that the per node capacity in an ad hoc network is of the order $O(\sqrt{n})$, which was considerably low. However, Jinyang et. al. [29] argued that it is possible to scale ad hoc networks if locality of traffic is maintained. This is because capacity decreases only if the expected path length increases. In [17], it was shown that mobility can actually increase the network capacity if routing uses path diversity.

Authors of [30] gray zones within which, data transmission hampered inspite of valid routing table entries. After having exposed this problem for AODV, they emphasized the importance of making routing decisions based on end-to-end link quality than on local decisions. In one of the first implementations of ad hoc routing protocols [47], DSDV was implemented over a link quality based routing metric. It was argued that minimum hop count is not the most effective metric as link quality was signficantly vary in a non-deterministic fashion in a wireless network. Limitations of hop count metric were also exposed in [24]. Authors also demonstrated that under a realistic setting, when the sources tended to be burtsy, addition of new nodes can actually improve network performance. This is because richer connectivity, provides increased opportunity to route around hot-spots. Using results of link-level measurements of an ad hoc network, authors of [12] proposed that neighbour abstraction is a poor approximation of reality as most node pairs that communicate well have intermediate loss rates. Authors in [3] propose and evaluate a multipath routing protocol, which builds maximally disjoint paths on demand. Through simulations they demonstrate the increases robustness provided by multipath routing. However, they use hop count as a routing metric; demerits of which are already exposed. In [27], a wireless manifold was proposed, which is a two dimensional surface whose geodesic distances accurately capture wireless signal propagation.

2.5 Related testbeds

Authors of [12] proposed ETX, a metric for high throughput. ExOR [7] is a path-based routing protocol that uses the ETX metric. Although ExOR has been able to achieve better performance than earlier protocols, it has a constraint of static topology. ExOR was implemented over RoofNet, a university-wide mesh testbed. Researchers in [13], developed another routing protocol for adhoc networks within which each node has more than one radio. This multiradio multihop routing protocol is designed over the ETX metric and again is only suitable for networks with static topology. A new TDMA-based MAC is evaluated in [41]. Authors of [34], proposed a framework for implementing MANET protocols.

2.6 Related applications: Delay tolerant networks and Sensor Networks

Khaled et. al. presented controlled flooding techniques [21] for large scale sparse mobile networks. Epidemic routing techniques are presented in [42], where networks may not have connected paths between source and destination at the same time of message transmission. Authors in [23] proposed a data-centric variation of controlled flooding. Authors of [48] focus on efficient message delivery in sparsely connected networks by introducing non-randomness in the movement of message carriers. [45] sheds light on connectivity analysis, neighbourhood management and routing for dense networks with low power radios and limited storage. Chhabra et. al. proposed a quality-based metric in [47] for routing in sensor networks and implemented DSDV over it. [5, 6] present a detailed review on the routing techniques used in sensor networks that are characterized by high node density, low power radios and limited storage capacity.

2.7 Why is a new routing metric required?

Several metrics have been proposed for routing in ad hoc networks. The shortest path metric for routing based on hop count was popular in early research on mobile ad hoc networks [26, 38, 37]. However, it was proven ineffective in [12]. End-to-end delay is another potential metric, but is undesirably sensitive to network load. As suggested in [12], a good metric should be independent of the network load as load balancing can be handled by separate algorithms. By far, the most effective metrics for ad hoc networks have been ETX [12] and its extensions [13]. However, these metrics do not take mobility into account. They are calculated on a per-link basis. If metrics like ETX are supposed to be used for mobile ad hoc networks then the underlying routing protocol has to propagate route metrics quickly enough, provided accurate link measurements are available. Both these assumptions are inaccurate considering the fact that these metrics are calculated on a per-link basis. We argue that transience can be efficiently handled by the system if the metric is based on end-to-end link measurements. The end-to-end approach ensures that the effect of node failures, node mobility and interference gets naturally factored into the metric. The new metric proposed by us called reachability is such a metric based on end-to-end link level measurements. It easily enables multipath-routing and also removes the need of maintaining symmetry across links. Additionally, for an N -node network, the storage requirement for path-based or link-based routing metrics (references) is bounded by $O(E)$, which could be $O(N^2)$ in MANETs, whereas reachability enables reliable routing in just $O(N)$ space.

CHAPTER III

DESIGN PRINCIPLES

This chapter elaborates on the design principles that guided our work. These design principles serve as a rationale behind the design and implementation of the flexible routing protocol. Our work began with only a few high-level guidelines. But as our system became mature after continuous iterative improvements, these guidelines evolved into concrete design principles. These design principles have been formulated by careful review of prior literature (see Chapter 2). Hence they also reflect the reasons behind the failure of existing systems in handling transience. In order to fully appreciate our design and implementation decisions, I strongly encourage the reader to carefully read this Chapter.

3.1 Use of commodity hardware and systems

A networking software is a complex system because it depends on several other systems and sub-systems. For example, consider the Transmission Control Protocol (TCP) as a network solution. TCP is present at the transport layer of the networking stack and its performance is directly affected by each of the layers below it, namely the network layer, MAC layer and the physical layer. Each of these layers is an independent system by itself. The actual hardware and the physical communication protocol (for e.g. Ethernet, WiFi) also affects the performance of TCP. The performance of TCP also depends upon security mechanisms present at layers below it (for e.g. IPSec). Thus as a networking solution, TCP is an extremely complex system because it affects and is affected by several other systems and sub-systems. One can easily surmise that even a slight change to TCP can have drastic repurcussions on the overall system performance. Unless there is a mechanism to extensively simulate and evaluate the changes beforehand, it is practically impossible to understand the effects of that change until they actually occur.

Secondly, changing systems that run on optimized hardware needs changes to the hardware itself. Consider the WiFi 802.11g MAC protocol as an example. If any solution directly changes the 802.11g protocol then there is a high chance that the hardware such as WiFi routers, which is optimized for 802.11g and which is used by billions of users, is rendered useless.

Due to these afore-mentioned problems and practical limitations in acceptance, we believe that changing the existing systems, i.e. in our case changing the 802.11 a/b/g MAC is not a suitable approach. We hence designed flexible routing as a new plug and play transparent layer into the networking stack. This approach offered us the following advantages:

1. It allowed efficient implementation in the kernel; kernel implementation is efficient because the number of context switches during packet forwarding are considerably reduced

2. It allowed the new layer to be transparent to the higher layers, allowing compatibility with existing network and transport protocols

3. It allowed the routing functionality to be MAC-based instead of IP-based, which significantly reduced the network configuration overhead

4. It allowed us to maintain the stock MAC, which made our solution more generic and hardware independent.

3.2 Throughput can be traded for reliability and usability

Multihop ad hoc wireless networks have not delivered on their promise. As detailed in Chapter 2 this is in part due to their capacity limitations. In the seminal work done by Gupta and Kumar [18], they showed that the capacity of multihop communication is bounded by $O(1/\sqrt{n})$. This was a significant result and shed light on the fact that scalability of wireless communication over multiple hops has strict capacity limits, especially for high bandwidth applications. Recent implementation efforts have achieved substantial throughput improvements at the expense of flexibility and reliability. Biswas and Morris [7] developed and evaluated an opportunistic protocol called ExOr that utilized the ETX metric [12] to achieve substantial throughput improvements over the traditional ad hoc routing protocols.

However, there are still two major problems:

1. The performance improvements are still not good enough to warrant real-life use, especially under transience for high throughput applications.

2. The performance improvements came at the cost of reliability and usability. For example, the ExOr protocol is designed as a link-state protocol for networks with static topologies only (mesh networks).

These problems have been a major hindrance in the real life deployment of ad hoc wireless networks. The motivation behind this design principle is that since the capacity of multihop wireless networks is inherently insufficient, there is no harm in trading throughput for higher levels of reliability, flexibility and usability, if doing so promises to serve some critical needs. We believe that by trading efficiency, it is possible to achieve the required reliability for handling transience and coming up with a practically feasible solution for low bandwidth applications such as disaster relief communication, sensor networks, etc.

3.3 Availability under eventual consistency

3.3.1 ACID properties

For distributed systems such as databases there are several desirable properties. These properties were formally proposed by Gray [?] and Haerder et. al. [20] in 1980s.

1. *Atomicity* - In a transaction involving two or more discrete pieces of information, either all of the pieces are committed or none are.

2. *Consistency* - The system should remain in consistent state with regard to any constraints. A transaction either creates a new and valid state of data, or, if any failure occurs, returns all data to its state before the transaction was started.

3. *Isolation* - Ensuring that the temporary state consequent to one group of actions is not visible to another group of actions occurring concurrently.

4. *Durability* - The successful completion of the group of actions results in a permanent change of state of the system.

Any relational database should satisfy these properties in order to function correctly. This approach works fine for relational databases. However, researchers from the systems area have long argued against using this approach for distributed data systems [15].

Eric Brewer conjectured that it is impossible for a distributed system to satisfy all three of - consistency, availability and partition tolerance (also referred to as *CAP properties*). The conjecture was formally proved by Gilbert and Lynch [16]. CAP properties are desirable in a distributed data system. The *CAP theorem* [8] states that it is impossible to satisfy CAP properties simultaneously in a system and a practical trade-off between them is essential for smooth and efficient functioning.

3.3.2 BASE properties

A system supporting ACID emphasizes consistency at the expense of partition tolerance and availability since it may become unavailable in the event of a partition occuring that causes transactions to fail. The term BASE was coined in [15]. BASE can be interpreted as - Basically Available, Soft state and Eventually consistent. The central idea is the system is always available - at the expense of inconsistency in the event of a partition occurring. Eventual consistency refers to the property that when the partition heals, the stale data will be updated to the fresh value and the system will be consistent once again. The ability of a BASE system to function in the event of a partition also increases the scalability of the system as a whole.

3.3.3 BASE properties for a routing algorithm

A routing algorithm is also a distributed data system. Particularly under transience, where node failures and node mobility are common, network and route partitions occur frequently. We argue that for the routing algorithm to handle transience and resulting route partitions efficiently, it should be designed keeping the BASE semantics in mind. The routing protocol should focus on high availability at the expense of consistency in topology information at individual nodes. The flexible routing protocol is designed by focusing on the BASE semantics.

- The *Effective Distance Table* (see Chapter 4) serves as an eventually consistent view of the entire network at individual nodes.

- By using eventually consistent topology information from the *EDT* the flexible routing algorithm ensures high availability under transience by taking a multipath routing approach.

CHAPTER IV

THE REACHABILITY METRIC FOR TRANSIENT ENVIRONMENTS

Metric is a property of a route in computer networking, consisting of any value used by the routing algorithms to determine whether one route should perform better than the other. One of the major challenges in the realization of MANETs has been the lack of a routing metric that effectively captures transience. Traditional metrics such as hop count, bandwidth, delay, etc. have already been proven ineffective. Although, newer metrics such as ETX [12] have worked well in static networks, they are ineffective in capturing transience like mobile nodes and failing nodes.

4.1 Desirable properties of a routing metric that aims to capture transience

We believe that a routing metric needs to have some desirable properties in order to successfully handle transience. A metric, which satisfies these properties, then empowers the routing protocol with tools to route packets successfully under varying degrees of transience. Reachability metric was designed in consideration of these properties. Our evaluation in Chapter 6 shows that reachability accurately captures transience and enables the flexible routing protocol to reliably route packets under varying degrees of transience.

4.1.1 End-to-end measurements

For a metric to capture transience, it should derive its value from end-to-end network measurements. This is particularly true for multihop communication. If a route spans multiple hops, then network conditions at each of the hops or intermediate nodes are important and should get appropriately captured or factored in the value of the routing metric. The metric should essentially empower the routing protocol to take more globally network aware routing decisions. The traditional approach of per-link measurements is observed not to work well.

4.1.2 Capture availability

We have argued before in Chapter 3 that the routing protocol should be highly available. It is essential for the routing metric to capture this aspect of availability in some way or the other, for empowering the routing protocol to take availability-aware routing decisions. Since the proposed flexible routing protocol aims for high availability with multipath routing, the routing metric reachability is designed to capture the effect of multipath communication. Herein lies the key difference between the ETX approach and our approach. ETX is a per link metric. Although ETX does capture end-to-end effect for a single path as the end-to-end ETX value is the sum of ETXs of the individual links, it does not, in any way naturally capture the affect of high availability, which the reachability metric does so efficiently.

4.1.3 Easy and bandwidth-efficient to calculate and maintain

Lastly, the routing metric should be extremely easy and bandwidth-efficient to calculate and maintain. Calculation of the routing metric values is a control overhead, hence should be as minimum as possible. This property especially becomes critical in transient situations to achieve a practical trade-off between the accuracy and efficiency of calculating and maintaining the routing metric values.

4.2 Reachability

4.2.1 Intuition

Our routing method is based on the notion of reachability, a directional metric, which captures the effects of transience in a single numerical value. Roughly speaking, it measures the end-to-end, multipath probability that a packet transmitted by a source node reaches the destination node. It is important to note that this probability should be over all possible paths and not any single path (unlike previous routing metrics, e.g., [12]).

In other words, reachability measures the maximum number of ways by which a packet transmitted by a source can reach the destination within a fixed number of hops. As this number is affected by physical obstructions, node failures and changing topology, we claim that reachability effectively captures transience.

Please note that reachability is an end-to-end metric. Reachability between two nodes is calculated only from the end-to-end network measurements conducted at those two nodes. The reachability calculation algorithm also ensures that the reachability value between two nodes captures the actual multipath availability between the two nodes. Lastly the algorithm itself is extremely efficient as it exploits the broadcast nature of the wireless channel to its advantage.

4.2.2 Definition

Definition. Reachability(A,B,T,L) of node B from node A is defined as the expected number of packet copies received by B for every packet originated at A and diffused in the network for at most L hops in time interval T.

4.3 Measuring reachability

4.3.1 Idea

We note that reachability is a directional metric. To understand how it is measured, consider a random node placement shown in figure 3 with SRC as the node of interest. We have to measure the reachability of all other nodes from SRC.

The intuition behind the algorithm is very simple. To measure reachabilities of all other nodes, the node SRC periodically floods the network with special control

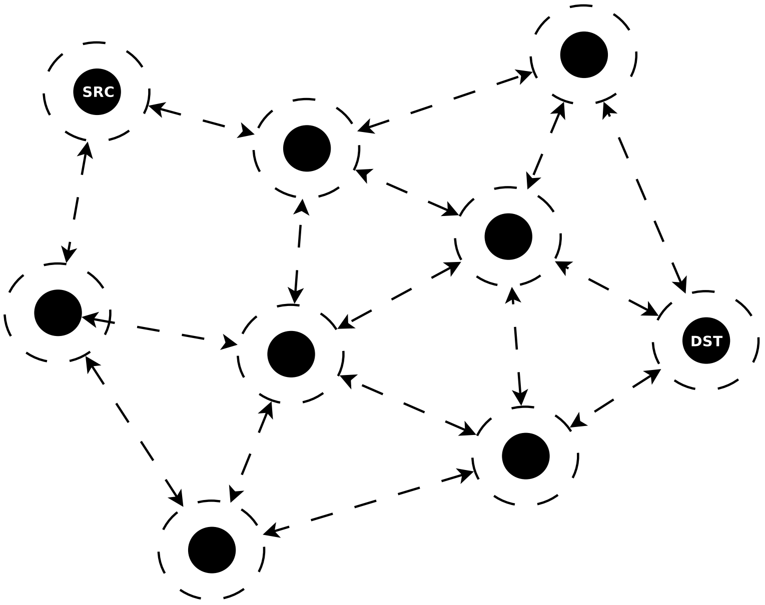


Figure 3: Measuring Reachability

packets called *heartbeat packets*. These heartbeat packets start with a fixed time-to-live (TTL) value at SRC and get diffused into the network until their TTL reduces to zero and they die down. Reachability of any node from SRC is measured by the fraction of the SRC's heartbeat packets that were received by that node per time period. The algorithm exploits the multi-access nature of the wireless channel and is extremely efficient than unicast flooding.

Reachability of node DST from node SRC is calculated as:

$$Reachability(R_{SRC.DST}) = R_{DST}/S_{SRC}$$

4.3.2 Mapping reachability to distance: Effective Distance

Traditionally in routing, 'distance' has been thought of as an indicator of closeness; lower the distance between two nodes, the closer they are and vice versa. Going with the same philosophy, we mapped reachability to a finite value, roughly its inverse. We call that value *Effective Distance*. Figure 4 shows the graph of effective distance

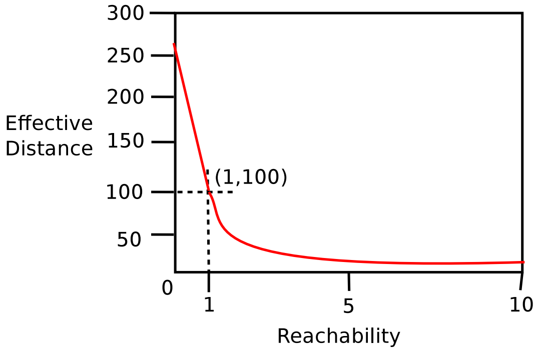


Figure 4: Mapping reachability to Effective Distance

as a function of reachability. This is the actual value of reachability used for routing.

$$ED = \begin{cases} \frac{100}{R} & \text{if } R > 1 \\ 255 - (155R) & \textit{otherwise} \end{cases}$$

4.3.3 Measurement Algorithm

Figure 5 shows the data structures that individual nodes maintain in order to compute reachabilities. Each node maintains three types of data - transmission statistics, reception statistics and a table of pairwise reachabilities.

4.3.3.1 Transmission Statistics Maintenance

Individual nodes periodically transmit special control packets called *heartbeat packets* destined to the broadcast MAC address. These packets then get diffused in the network. We call each time period a *session*. *txSession* refers to the current session number and *txCount* is the count of packets transmitted by a node in *txSession*.

4.3.3.2 Reception Statistics Maintenance

A table is stored at each node, which contains the count of heartbeat packets received from other nodes along with their session numbers. This helps the other nodes calculate the reachability of that node. While transmitting heartbeat packets (see

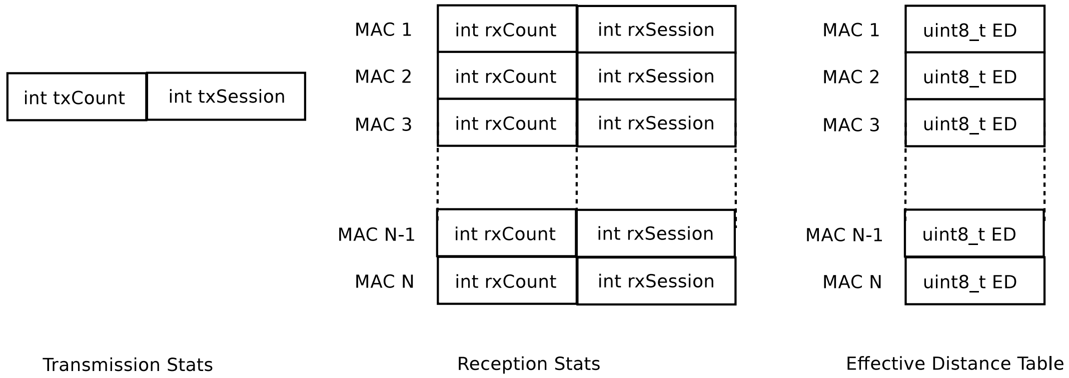


Figure 5: Data structures for reachability measurement

above section 4.3.3.1), this information is piggybacked onto them, so as to enable other nodes glean relevant reception information and calculate reachabilities.

4.3.3.3 Calculation of Reachabilities

By using their transmission statistics (Section 4.3.3.1) and reception statistics gleaned from received heartbeats (Section 4.3.3.2), individual nodes calculate effective distances of other nodes using the formula given in Section 4.3.2. These effective distances are stored in a table called the *Effective Distance Table* or *EDT*. EDT serves as a compact ($O(n)$ size) view of the entire network and is sufficient for making routing or forwarding decisions. For details on routing please read Chapter 5.

4.4 Why reachability?

As mentioned earlier, end-to-end measurements, ability to capture availability and bandwidth efficient maintenance are three important characteristics of a routing metric for successfully capturing transience and helping the routing protocol sucessfully route packets. It is clear from the previous literature that traditional metrics like hop count, bandwidth, delay, etc. do not capture transience accurately. Recently proposed per-link metrics like ETX [12] and its variants, work well for static networking topologies, but do not capture the effect of node mobility. Although they do allow one to perform end-to-end estimations, they are not based on end-to-end measurements per se. Moreover these metrics don't enable a compact and easily maintenable network representation that is critical to handle transience. Lastly, they are designed for single path routing and do not capture availability, which is also important.

On the other hand, reachability is based on truly end-to-end measurements that give an accurate picture of the network state. Reachabilities can be easily measured by a bandwidth-efficient method, which as our evaluations show, works under varying degrees of transience. The reachability metric also enables a compact network representation, called effective distance table, which is an eventually consistent view of pairwise reachabilities. In Chapter 6, we present detailed evaluation results which show that reachability successfully captures:

- the effect of increased connectivity as the network scales
- the effect of degraded connectivity as node failures happen
- the effect of mobile nodes

CHAPTER V

FLEXIBLE ROUTING PROTOCOL

The flexible routing protocol is a new routing protocol, which is based on the reachability metric. Flexible routing is a multipath routing protocol that uses pairwise reachabilities to reliably deliver packets under varying degrees of transience. It trades throughput to achieve the required reliability in communication. This work borrows many concepts from the early work done by Ashwin and Santosh [34]. They proposed a framework called MyMANET for implementing mobile ad hoc routing protocols. It used Virtual Distance as a routing metric, which was based on end-to-end packet loss. A primitive routing protocol based on Virtual Distance was also implemented. Reachability and the flexible routing protocol emerged and were concretized after extending and re-engineering MyMANET many times.

Maintaining paths explicitly is not practical under transience. Hence the core routing decision for flexible routing is *'Whether or not to forward'* instead of *'Which node to forward to?'*. Each node maintains a compact table of pairwise reachabilities (EDT) computed from receiving heartbeat packets (Section 4.3.1), and uses these to selectively forward data packets, effectively pruning a flood tree. Although paths are not being created or maintained, this opportunistic approach ensures that the packets end up travelling along multiple available paths towards the destination. Flexible routing could be considered in the same spirit as probabilistic forwarding techniques except for two important differences - (1) flexible routing performs packet forwarding with a "network-aware" probability, which is governed by the reachability metric and (2) the forwarding mechanism is much more efficient than unicast flooding. In other words, the routing algorithm ensures that packets on the network are forwarded by only those nodes that are likely to increase the chances of the packets reaching the destination.

5.1 Design Goals

As mentioned earlier in Chapter **??**, one finds many scenarios in real life that lack a dependable and affordable communication solution. Communication in the aftermath of disasters, communication in remote-resource constrained areas, communication in oil and natural gas sites, on-ship maritime communication, on-field communication for media personnel, communication during trekking, mountaineering and archeological expeditions, wireless sensor networks, etc. are some examples. There scenarios have two main problems - (1) Transience and (2) Lack of infrastructure. Flexible routing aims to satisfy the needs of such scenarios.

5.1.1 Reliable communication in transience

How to communicate reliably in transience, has always been a very hard problem to solve for the research community. We argue that the primary reason for the absence of any solution is lack of a routing metric that efficiently and accurately captures transience. The second subsequent reason is lack of a routing protocol that naturally provides fault-tolerant communication in varying degrees of transience. We refer to transience by - (1) frequently changing topology due to mobile nodes, (2) node failures and new nodes joining the network, (3) changing physical obstructions and (4) internal and external interference. Such transient conditions are frequent in the scenarios mentioned above. No existing routing metric is able to accurately capture these conditions. Moreover, this also renders existing routing protocols incapable of providing reliable connectivity in transience.

5.1.2 Minimum use of infrastructure

Secondly, lack of infrastructure mostly due to lack of sufficient resources or difficulty in establishing infrastructure, is another characteristic feature of these scenarios. Most popular wireless communication solutions are infrastructure-based and hence not applicable in such scenarios. For example, in a remote rural village in a developing country, establishing a GSM base station is infeasible due to two reasons - scarcity of resources in that village and the possiblility of lower returns on investment due to lower user density.

5.2 Design

We argue that the design goals of reliability under transience and use of minimum infrastructure must be satisfied for any solution to work in the scenarios mentioned above. The design of flexible routing protocol is backed by some principles that have been elaborated in Chapter 3. When we began working on the flexible routing protocol, we had to make certain design decisions in order to empower the protocol in achieving the collective goal of reliability and minimum infrastructure-usage under transience.

5.2.1 No functional hierarchy

We argue that in order to successfully handle transience, reliability should be fundamentally built into the system. The easiest way to fundamentally achieve high reliability is to use a completely distributed routing approach. Naturally it comes at the cost of performance. But as mentioned earlier, increased reachability helps solve some critical needs, which amortizes the cost of reduced throughput.

Contrary to traditional wireless networks such as cellular networks [32, 43] and WiFi networks, flexible routing does not employ any functional hierarchy amongst nodes in the network. Nodes may be based on different hardware platforms, but they share the same piece of software and capabilities. This makes the network naturally fault-tolerant and robust to node failures and movements. This approach is also easy to scale, provided the system designed is basically available and eventually consistent.

5.2.2 Proactive routing

In proactive routing individual nodes periodically update the topology and route information so that fresh information is available before any data transfer starts. In the reactive approach, the topology and route information is updated on demand just before the data transfer starts. As elaborated in Chapter 2, the reactive approach is not suitable for transient networks. Following are the reasons:

- For large networks, building up topology information on demand is time consuming. This problem becomes more acute for multipath routing protocols. In other words, the reactive approach does not scale well.

- For conditions of high transience, updates to the topology information have be more frequent. Thus even if topology information is populated on demand, its maintenance has to be proactive as long as the communication is happening.

As we show later in the evaluation (Chapter 6), our proactive approach enables reliable communication under varying degrees of transience and is extremely bandwidth efficient.

5.2.3 Multipath routing without explicit maintenance of paths

The core routing decision format for path-based routing protocols is - *'Which node should I forward the packet to?'*. Such a decision allows the sender to pre-compute or compute the path on demand to any destination. This approach works well in networks that are free of transience. But under transience, maintenance of such precomputed or on-demand paths is extremely difficult. We realized this and decided to do away with the idea of establishing or maintaining any paths.

The decision taken by the flexible routing algorithm is - *'Whether to forward or not?'*. The core routing mechanism is very simple. The source node transmits a packet on the network. Nodes that receive that packet make a forwarding decision of *'Whether or not to forward?'*. Based on the forwarding decision algorithm, a subset of nodes forward the packet further. The forwarding decision algorithm ensures that the packet is forwarded by only those nodes that are likely to increase the chances of the packets reaching their destination. This forwarding continues until the packets eventually reach their destination or die down. Although paths are not being created or maintained, this approach ensures that the packets end up travelling along multiple paths towards the destination.

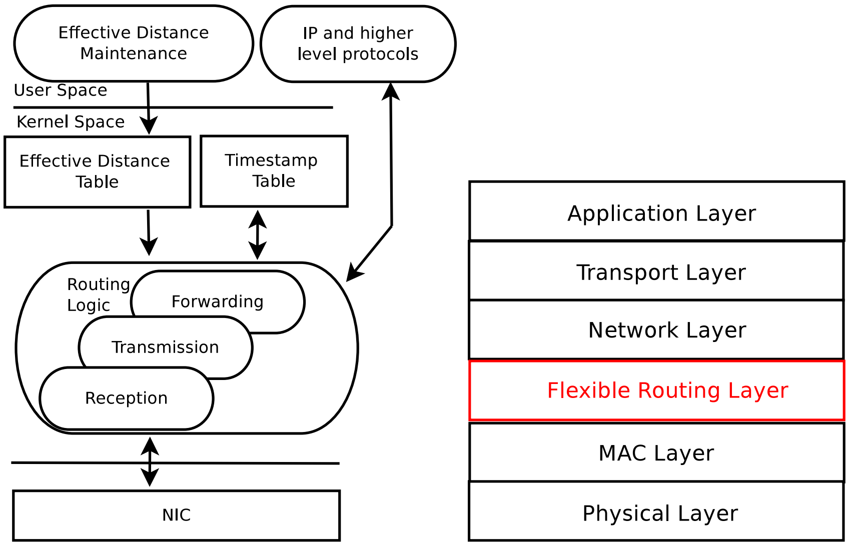
Multipath routing offers the following advantages:

- **High availability** - Even during conditions of high transience, one or more of the multiple available paths lead the packets to their destination.

- **Fault-tolerance** - Due to multipath routing, failure of a few nodes does not degrade the performance of an active flow.

5.2.4 Tradeoff: Reliability versus Throughput

'Reliability versus throughput' is a critical tradeoff in the design of the routing protocol. As argued in earlier chapters, reliability is more important than throughput in the context of handling transience. We achieve highly reliable communication by utilizing highly available routing techniques like multipath routing. This of course comes at a cost of reduction in throughput. But we argue that the cost of reduced throughput gets amortized by fact that increased reachability helps satisfy critical needs in scenarios such as communication in disaster relief, communication in remote rural areas, remote sensor networks, etc. For these scenarios, easy and rapid establishment of baseline connectivity in highly transient environments is priority as against high throughput.



(a) Flexible routing architecture (b) Layer 2.5 implementation

Figure 6: Flexible Routing Architecture

5.3 Architecture

We implemented the flexible routing protocol by extending and re-engineering the mobile ad hoc networking framework called MyMANET, proposed by Ashwin and Santosh [34]. Figure 6(a) shows the architecture. It is evident from the architecture that some components of the routing protocol reside in the kernel, while some at the user level. Figure 7 shows the high level functional block diagram. The entire routing functionality can be broadly divided into two components:

1. Effective distance maintenance (EDM)

2. Routing

Effective distance maintenance (EDM) functionality ensures that the Effective distance table (EDT) is regularly updated. EDT is a table of pairwise reachabilities maintained at every individual node. It is implemented in the user space. Routing functionality, which resides inside the kernel, uses data from EDT to make routing decisions. The relation between EDM and routing could be thought of as a producer

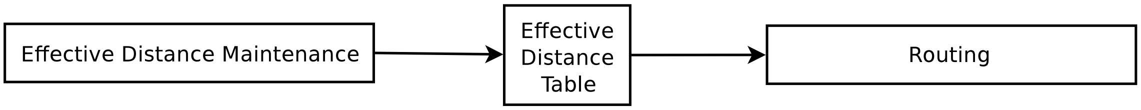


Figure 7: Functional block diagram of the routing protocol

consumer relation where EDM produces data and routing consumes it. Packets on the network carry an additional header (shown in figure 8). The routing kernel module intercepts packets between the network and the MAC layer to carry out header insertions, modifications or deletions and is hence referred to as a layer 2.5 implementation.

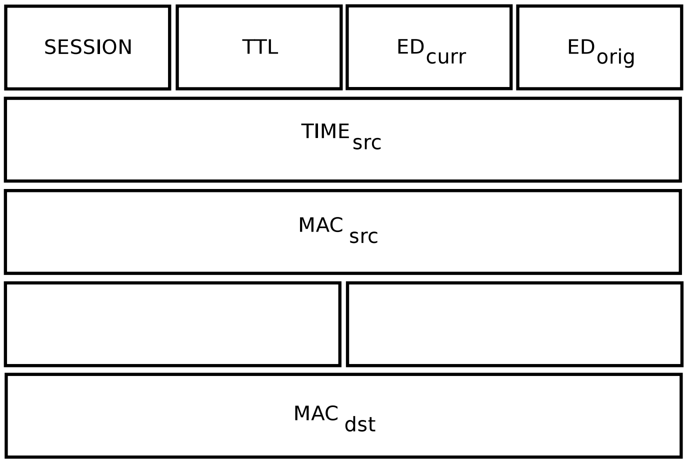


Figure 8: Flexible routing header

Figure 9 shows the data structures used in routing. Data structures related to effective distance maintenance are present in user space whereas data structures directly referenced by routing are implemented in the kernel space. Effective distance maintenance functionality proactively keeps the EDT updated using the heartbeating mechanism. Since this mechanism is a control overhead and is not performance critical, it can be implemented in the user space. Hence the transmission and reception statistics are implemented in the user space. On the other hand, EDT and the timestamp table are frequently referenced by the layer 2.5 kernel module. Since the kernel

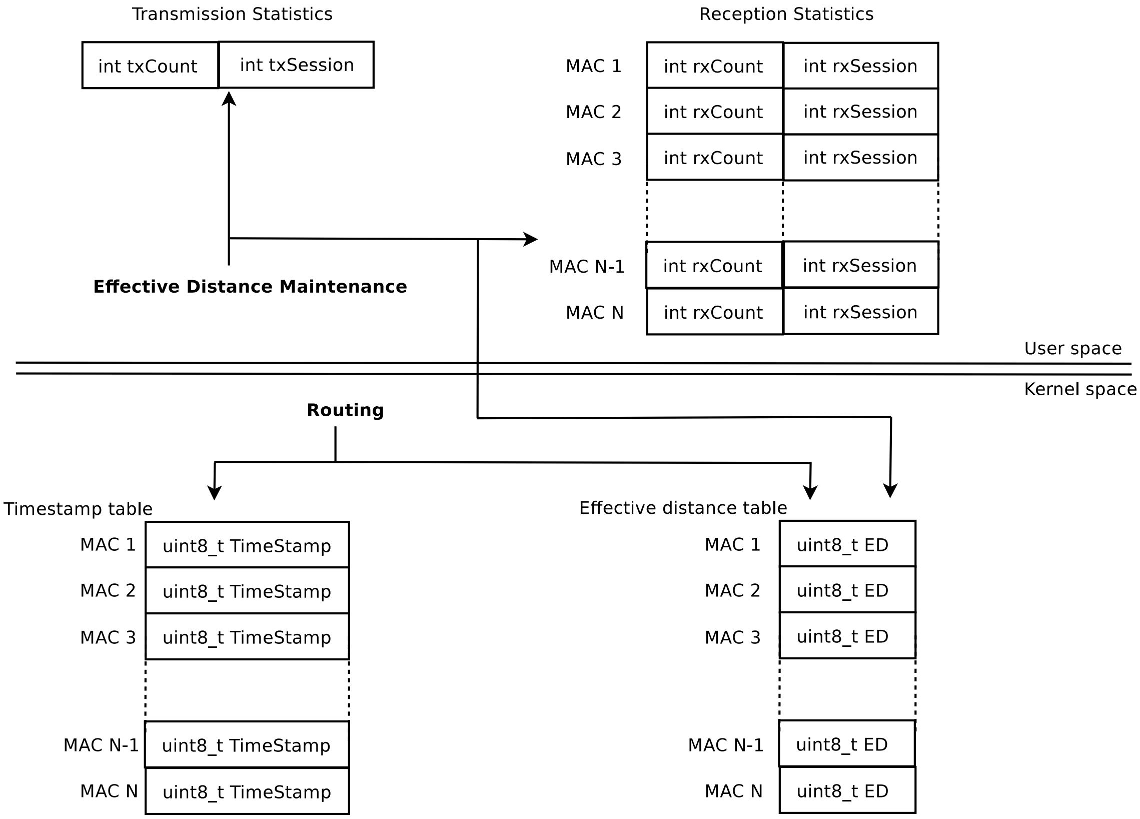


Figure 9: Data structures used

module is a performance critical software component, EDT and the timestamp table are implemented in kernel space for eliminating the context switching overheads.

5.4 Effective distance maintenance

EDT stores the effective distances of all nodes. To calculate and update effective distances, it is necessary to first calculate the reachability values. Transmission and reception statistics (see figure 9) are used to calculate reachabilities and update the EDT. Transmission statistics contains the count of per session transmitted heartbeats. Reception statistics contain the counts of heartbeats received from other nodes per their respective sessions. Transmitted heartbeats are piggybacked with reception statistics information in order to help other nodes in their respective reachability calculations. A node calculates the reachabilities of other nodes and updates the EDT by using information from its transmission statistics and information gleaned from received heartbeats. The method for calculating reachability and effective distances is exemplified in Chapter 4.

5.5 Routing

The routing component is implemented at layer 2.5 as a kernel module. As shown in the architecture diagram 6(a), it performs three main functions:

1. Packet transmission

2. Packet forwarding

3. Packet reception

The source node transmits the data packet on the network. Nodes that receive that packet make a forwarding decision of *'Whether or not to forward?'*. Based on the forwarding decision algorithm, a subset of nodes forward the packet further. The forwarding decision ensures that the packet is forwarded by only those nodes that are likely to increase the chances of the packet reaching its destination.

5.5.1 Packet transmission

1. At the source of the transmission, all out-going packets are trapped by the kernel module into the kernel just before they are delivered to the MAC for transmission.

2. The kernel module of the source then cooks a new header as shown in figure 8. It fills the header fields as shown below:

   - $SESSION$ (write-once) - the current session number from the transmission statistics (see figure 9).
   - TTL (updated by forwarding nodes) - the time-to-live field. Usually it is set to a fixed value. We use 4 as the TTL value during our experiments.
   - ED_{CURR} (updated by forwarding nodes) - the effective distance of the destination node from the current node (the source node in this case).
   - ED_{ORIG} (write-once) - the effective distance of the destination node from the source node (the source node in this case).
   - $TIME_{SRC}$ (write-once) - timestamp at the source node, useful for sequencing and identifying duplicate packets at intermediate nodes.
   - MAC_{DST} (write-once) - MAC address of the source node. It copies this field from the destination of the MAC address.
   - MAC_{SRC} (write-once) - MAC address of the source node.

3. The kernel module of the source then inserts the newly cooked header into the packet between the network and the MAC headers.

4. Finally it modifies the destination node of the MAC address, and sets it to the broadcast MAC address *FF:FF:FF:FF:FF:FF* and hands it over to the MAC for further transmission.

5.5.2 Packet reception

All nodes that are in direct range of the source are able to receive the packet. Each of these nodes first checks if it is the final destination of the packet. A node can perform this check by comparing its MAC address with the MAC_{DST} field in the packet. If a node is the final destination it consumes the packet as follows:

1. The received packet is trapped by the kernel module of the receiver node before it is handed over to the network layer for further transmission.

2. It then compares its MAC address with the MAC_{DST} field in the flexible routing header. If equal, then the receiver node is the final destination of the packet. Else it is not.

3. If the receiver node is the final destination, it strips the flexible routing header and hands the packer over to the network layer for further processing.

4. Else, it takes the forwarding decision of *'Whether or not to forward?'*.

5.5.3 Packet forwarding

As mentioned in the above section, a receiver node becomes a prospective intermediate forwarding node if it is not the final destination of the packet.

1. The received packet is trapped by the kernel module of the receiver node before it is handed over to the network layer for further transmission.

2. It then compares its MAC address with the MAC_{DST} field in the flexible routing header. If equal, then the receiver node is the final destination of the packet. Else it is not.

3. If the receiver node is the final destination, it strips the flexible routing header and hands the packer over to the network layer for further processing.

4. If the receiver node is not the final destination, it has to take the forwarding decision of *'Whether or not to forward?'*.

5. The kernel module decides to forward the packet only if the effective distance of the final destination in its EDT is within a threshold α of the ED_{CURR} in the packet. This check ensures that the receiver node indeed increases the chances of the packet reaching its destination. This check is called the *reachability improvement check* and α serves as its tolerance parameter.

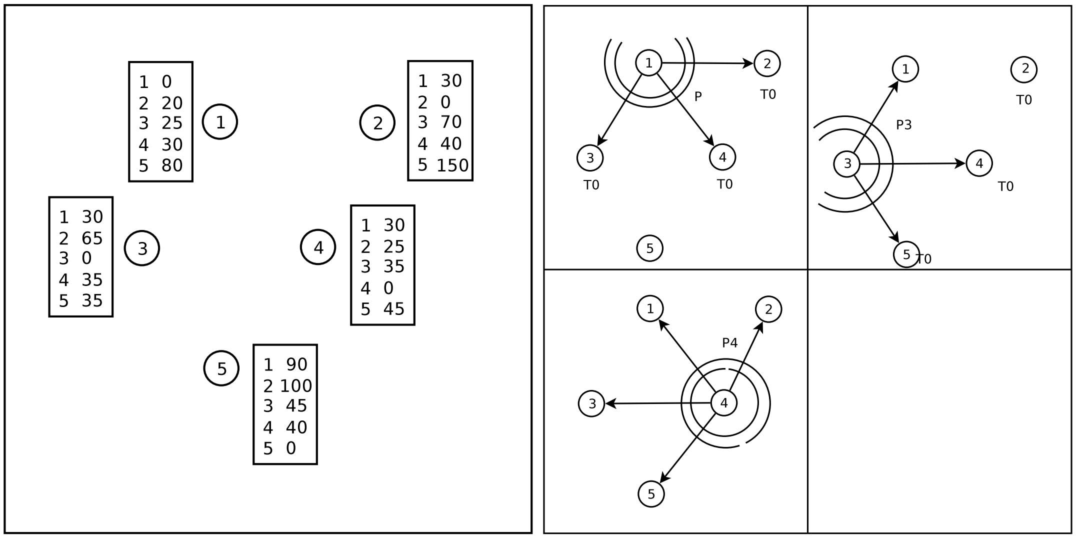
6. Timestamp table (see figure 9) stores the timestamp of the most recent packet received from every node and is used to identify duplicates is used to identify are forwarded only if ED_{SRC} is greater than the effective distance of of the destination node in the receiver's EDT by β at least.

7. Once the kernel module of a receiver node decides to forward the packet, it updates the ED_{CURR} field in the packet header with the effective distance from its EDT, decrements the TTL field in the packet header and hands it over to the MAC for retransmission.

5.6 Illustration

This section highlights the key aspects of the routing algorithm through an example. Consider a random node placement shown in 10(a). Assume *Node 1* to be the source node and *Node 5* to be the destination node. *Node 1* wishes to transmit a packet P to *Node 5*. We shall now see how it is routed. 10(b) shows the key forwarding decisions. Assume $\alpha = 30$ and $\beta = 20$.

- *Node 1* transmits the packet P on the network. Destination MAC address of P is set as *FF:FF:FF:FF:FF:FF*. Note that P has the Flexible Routing header between MAC and Network Layer headers. The header fields have values as - [$ED_{SRC} = 80$, $ED_{CURR} = 80$, $TTL = 4$, MAC_{SRC} = MAC of *node 1*, MAC_{DST}



(a) Topology

(b) Illustration stages

Figure 10: Flexible Routing Illustration

= MAC of *node 5*, $TIME_{SRC}$ = Current time at *Node 1*, *Session* = Session number of *node 1*].

- Since nodes 2, 3 and 4 are in direct range of *node 1*, they receive P. P is a new packet, thus all of them update their Timestamp Tables with the $TIME_{SRC}$ value in the packet header. Reachability improvement check is satisfied at nodes 3 and 4. Hence a packet copy of P is further forwarded by nodes 3 and 4 each. *Node 2* drops P due to the failure of reachability improvement check.

- Flexible Routing header in the packet forwarded by *node 3* (say $P3$) - (ED_{SRC} = 80, ED_{CURR} = 35, TTL = 3, MAC_{SRC} = MAC of *node 1*, MAC_{DST} = MAC of *node 5*, $TIME_{SRC}$ = Current time at *Node 1*, *Session* = Session ID of *node 1*).

- Flexible Routing header in the packet forwarded by *node 4* (say $P4$) - (ED_{SRC} = 80, ED_{CURR} = 45, TTL = 3, MAC_{SRC} = MAC of *node 1*, MAC_{DST} = MAC of *node 5*, $TIME_{SRC}$ = Current time at *Node 1*, *Session* = Session ID of *node 1*).

- Packet $P3$, which is forwarded by *node 3*, reaches *node 1, node 4 and node 5. Node 1* drops $P3$ as it is the source. *Node 5* consumes $P3$, as it is the destination node. $P3$ is a duplicate packet for *Node 4* as it has already seen P before. Since ED_{DST} is not greater than ED_{SRC} field in the packet by β or more, *Node 4* further forwards the duplicate packet.

CHAPTER VI

EVALUATION

This chapter presents a detailed evaluation of the effective distance metric and the flexible fouting protocol. Section 6.3 evaluates the network behaviour as it scales. Section 6.3 evaluates the performance of the network under node failures. In Section 6.5, we evaluate the network under mobility. Section 6.4 presents throughput measurements. Lastly in Section 6.7, we describe some mechanisms for controlling excessive redundancy. Each section presents some hypotheses and supports them with the requisite experimental evidence. With strong support from each of these hypotheses we then claim that dffective distance metric successfully handles transience and flexible routing uses effective distance to reliably route packets in transient networks.

6.1 Evaluation metrics

We first define two new metrics which were used for evaluation - Connectivity and Flow capacity. *Connectivity* relates to the network as a whole and captures how strongly are nodes connected to each other. *Flow capacity* relates to a traffic flow. For any pair of communicating nodes, flow capacity aims to capture the end-to-end capacity of the flow. We also used *reliability*, which was proposed by Karger et. al in [28].

6.1.1 Connectivity

Connectivity intends to measure how strongly are the nodes in a network connected to each other. *Node A* can be considered strongly connected to *Node B*, if the reachability of *Node A* is high from *Node B*. In other words, *Node A* is strongly connected to *Node B*, if the effective distance of *Node A* from *Node B* is less than some empirically

Table 1: Section-wise Hypothesis

Section	Hypotheses
Scalability	(1)Reachability captures the phenomenon that connectivity of the network increases as the network scales (2)Reachability captures the effect of physical obstructions (3)Flexible Routing improves end-to-end flow capacity and packet loss as the network scales
Node-Failures	(4)Reachability captures the effect of node failures (5)Flexible Routing successfully handles node failures (6)Flexible Routing ensures that a flow remains unaffected by removal of nodes that are not a part of it
Mobility	(7)Reachability captures the effect of mobility (8)Flexible Routing successfully handles mobility
Redundancy-Control	(9)Excessive Redundancy can be controlled in Flexible Routing

defined value that represents the threshold effective distance for strong connectivity. We can extend this idea and say that the entire network is strongly connected if all node pairs are strongly connected to each other. Connectivity essentially measures how far is a network from being strongly connected.

Definition (Connectivity) For a given network, let M denote the number of node pairs that are strongly connected and N denote the total number of node pairs. Connectivity (C) is then represented as $C = M/N$. Node B is said to be strongly connected to node A if effective distance of B from A is less than a threshold effective distance ED_{TH}, which is empirically determined

6.1.2 Flow capacity

Flexible routing is a multipath routing protocol, which allows packets to travel on diverse paths before they reach the destination. Since reachability is the only metric used by flexible routing, capacity of the diversified traffic flow is governed by the reachabilies of the nodes in the network. By being more specific we can say that the theoretical end-to-end capacity of any flow is decided by the reachabilities of all node pairs that constitute the edges of that flow. We define flow capacity as follows:

Definition (Flow Capacity) For a given network with a traffic flow F, let S denote the source node and T denote the destination node of the traffic flow F. Construct a weighted graph $G(V, E)$, such that the set of vertices V is the same as that of the network and E contains all the edges of the network that are a part of the traffic flow F . Each edge E_{ij} will have capacity C_{ij}, which is the reachability of node i *from* node j. *Flow capacity is then represented by the minimum $S - T$ cut of G.*

6.1.3 Reliability

Reliability was defined as the fraction of node pairs that remain connected when each node fails independently with some probability. We have used this metric in the Section 6.5.

6.2 Test Environment

All experiments were conducted in a university building. For minimizing external interference, they were conducted at night. Laptops and 802.11g WiFi routers (adhoc mode) with a 3dBi external antenna were used as test equipments. They were loaded with flexible routing software. Laptops had Ubuntu 10.04 as their operating system, whereas the WiFi routers were based on OpenWrt. Flexible Routing software was designed as a self-configuring system. To create or join a mobile ad hoc network,

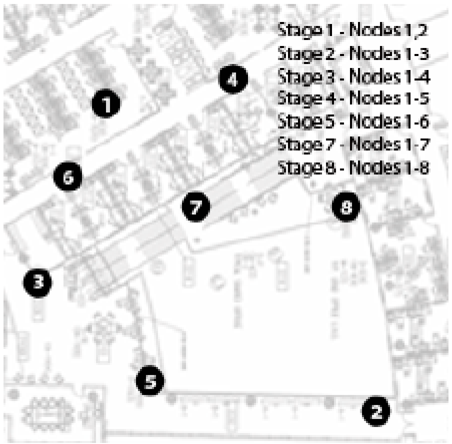
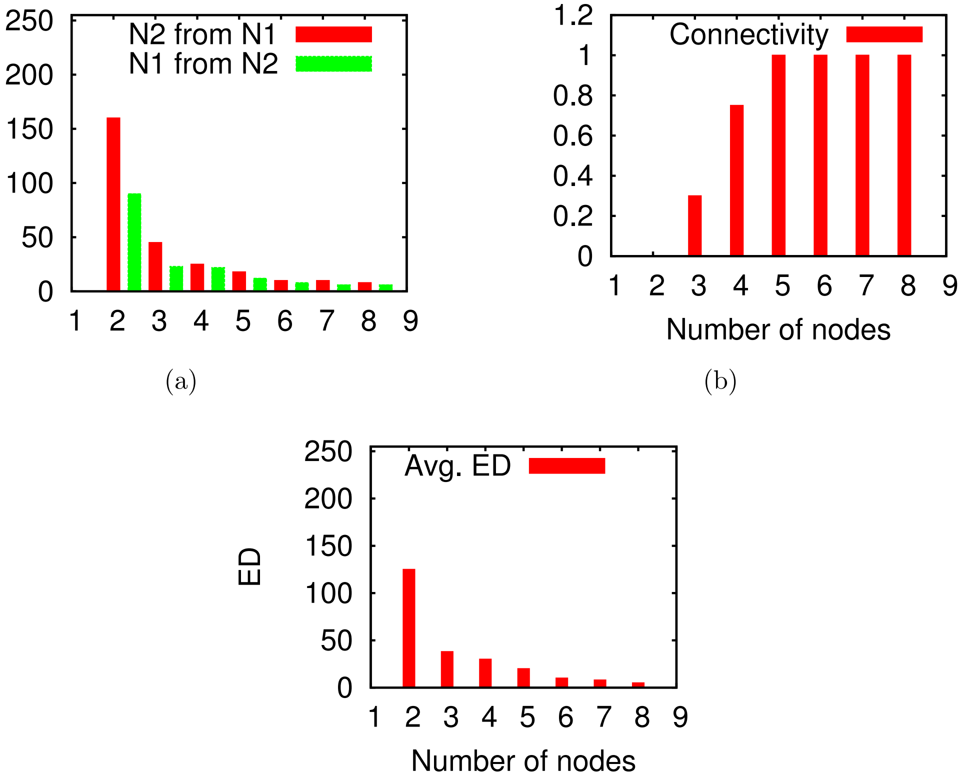


Figure 11: Scalability experiment setup

laptops just needed to execute the run script that loaded the software and auto-configured their IPv4 addresses. Routers were designed to create/join the network automatically after post-boot self configuration (i.e. after they were switched ON). The setup time was less than 10 minutes for each experiment described in this Section. Extensive data logging and traffic tracing mechanisms were built into the design of the system (with a run-time flag for enabling or disabling them). WiFi routers did not log any data because of their stringent memory requirements (8MB flash and 32 MB RAM).

6.3 Scaling the network with reachability and flexible routing

We conducted two experiments to see how the network scales. Both the experiments followed the experimental setup shown in Figure 11. The experiments were conducted in stages. The first stage consisted of two nodes that were kept far apart with several physical obstacles between them. Between the two first-stage nodes, new nodes were added in subsequent stages at randomly selected positions. Seventh stage was the last stage and consisted of eight nodes.



(a)

(b)

(c)

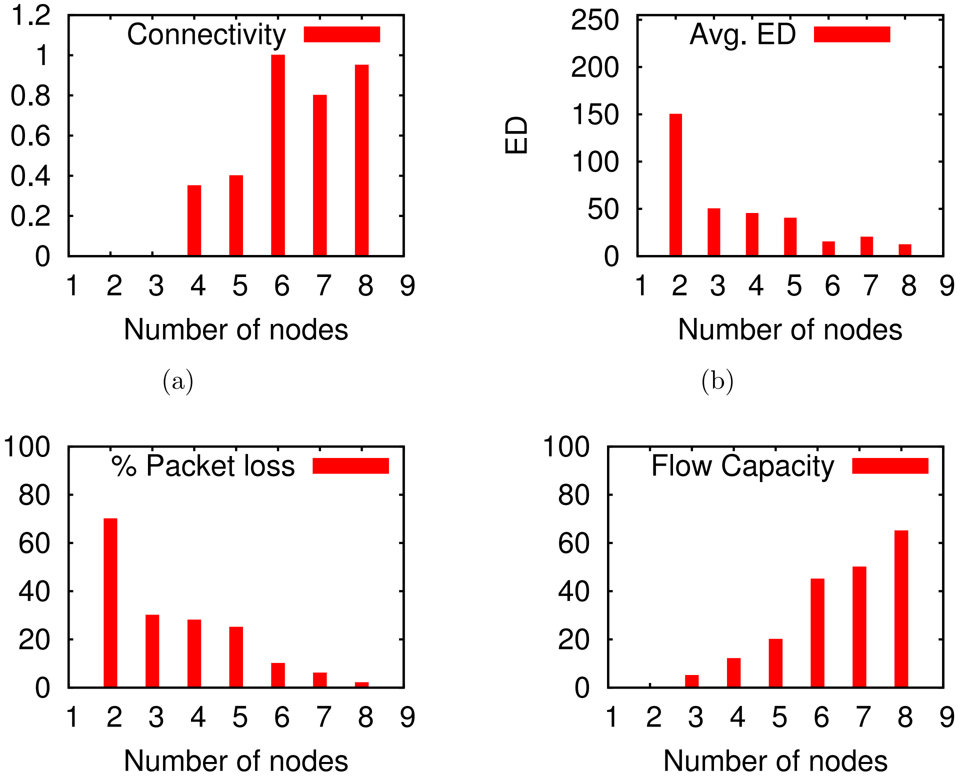
Figure 12: Scalability experiment 1

6.3.1 Experiment 1

The first experiment consisted of measuring effective distances of all node pairs in each stage. In the first stage that had just two nodes, the effect of physical obstructions and the geodesic separation between the two nodes was the most pronounced. This was reflected in very high effective distances between nodes 1 and 2 (see Figure 12(a)). As new nodes came in, the effect of physical obstructions started diminishing. This is because addition of new nodes resulted into new alternate paths that improved the connectivity of nodes 1 and 2. This diminishing effect is evident by the decreasing trend of effective distances between nodes 1 and 2 in Figure (see Figure 12(a)). Finally in the last stage the effect of physical obstructions was left to its minimum and correspondingly the effective distance values in Figure 9(a) were also at the minimum. Thus Hypothesis 2 is qualitatively supported by this experiment. Quantitative evidence for Hypothesis 1 can be seen in Figure 12(b), which shows that connectivity of the network increased as new nodes got added to the network. Figure 12(c) shows that average effective distance per node pair decreased which in turn caused the connectivity to increase. Figure 12(b) can be viewed as a fine-grained perspective on increasing connectivity.

6.3.2 Experiment 2

In the second experiment, a traffic flow was maintained in each stage between node 2 (source) and node 1 (destination). Traffic consisted of ping packets with interval set to 1 second. Figure 13 shows the results. Figure 13(c) shows that the percentage end-to-end packet loss between node 1 and node 2 reduced as the network increased in size. Figure 13(c) shows the increasing trend of connectivity. The increasing connectivity basically increased the flow capacity as seen in Figure 13(d). Increased flow capacity resulted into reduced losses. With support from Hypothesis 1 and evidence from figures 13(d), 13(c), 13(a) Hypothesis 3 can be clearly inferred. According to



(a)

(b)

(c)

(d)

Figure 13: Scalability experiment 2

Hypothesis 2, connectivity increased as the network grew. This meant that all nodes came close to each other in terms of end-to-end reachability (i.e. average effective distance per node pair decreased).

6.4 Failures

We conducted two experiments to see how the network performed under node failures. Both experiments followed the experimental setup shown in Figure 14. The experiments were conducted in stages. The first stage consisted of eight nodes as shown in the figure 14. In every subsequent stage, one randomly selected node was removed from the network (other than nodes 1 and 2). Seventh stage was the last stage and consisted of only two nodes (node 1 and node 2).

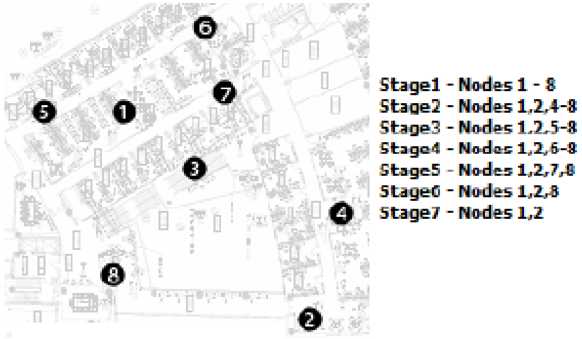


Figure 14: Node failures experiment setup

6.4.1 Experiment 3

This experiment consisted of measuring effective distances of all node pairs in each stage. As seen from figures 15(a) and 15(b), effective distances between node 1 and node 2 were lowest in the first stage. As the nodes started failing one per stage, effective distances began increasing due to the reduction in connectivity. Connectivity was reduced because removal of nodes from the network meant the removal of paths that connected nodes 1 and 2. In the last stage, there were no intermediate nodes left.

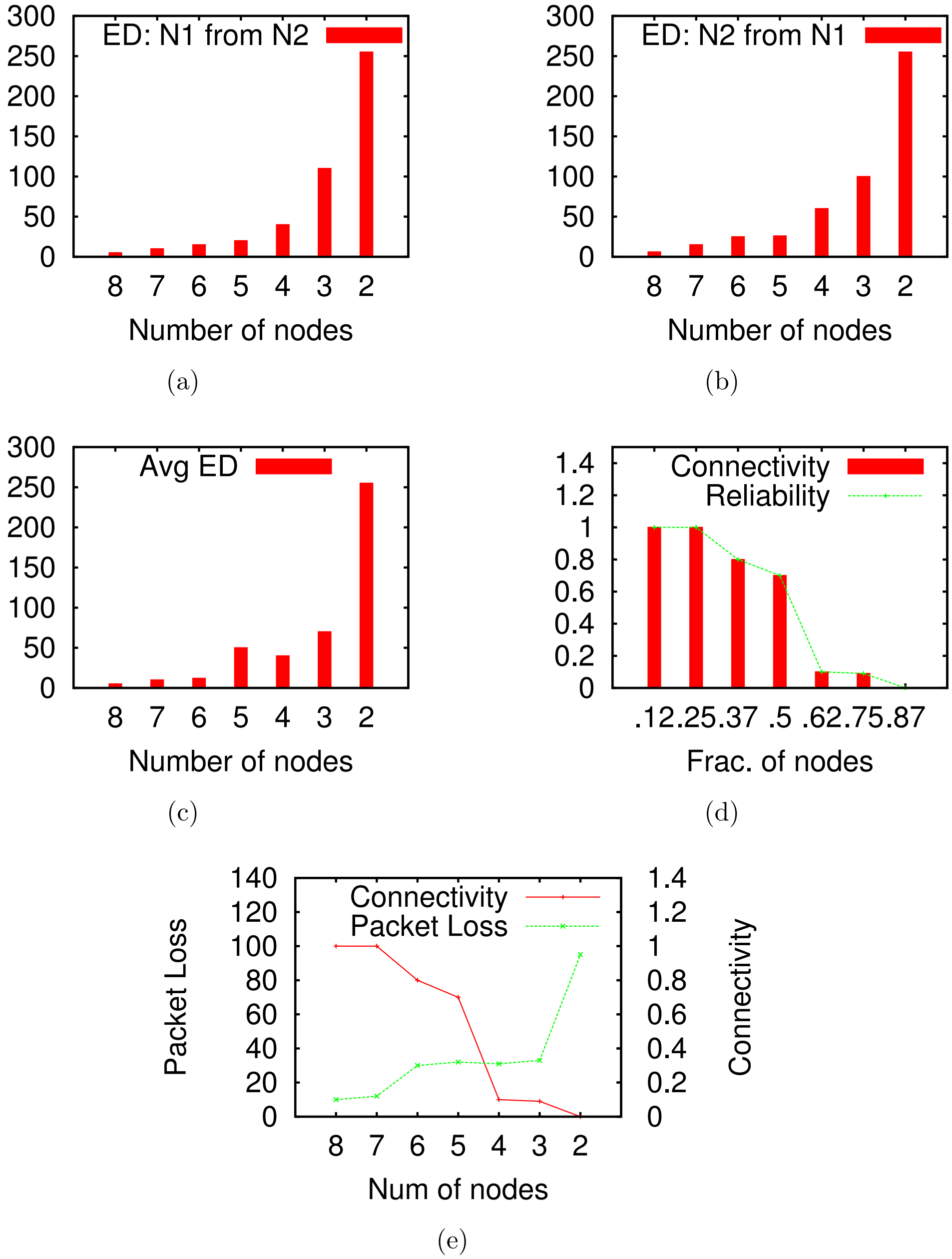
Nodes 1 and 2 were completely out of each others range and had maximum effective distance (255).

Thus the effect of node failures got accurately reflected in the values of effective distances i.e. in reachabilities (Hypothesis 4). Figures 15(d) and 15(c) show that connectivity of the network decreased and average effective distance per node pair increased as a result of node failures. Figures 15(d) also shows the reliability curve.

6.4.2 Experiment 4

In this experiment, a traffic flow was maintained in each stage between node 2 (source) and node 1 (destination). Traffic consisted of ping packets with interval set to 1 second. Figure 15(e) shows the results. Figure 15(e) shows that the end-to-end packet loss of the flow increased as nodes failed. This is because of the reduced connectivity as a result of node failures. The reduction in connectivity caused the reduction in flow capacity (not shown here due to space constraints) and ultimately led to the increase in packet losses. However, inspite of the failing nodes the flow was maintained until the last stage for very low values of connectivity. The flow was broken only in the last stage because the communicating nodes went out of each other's range as there was no intermediate node left. Hence we conclude that flexible routing is able to maintain traffic flow in presence of failing nodes (Hypothesis 5).

Evidence for Hypothesis 6 can be found by observing the stages 4 and 5 of figure 15(e). Nodes 5 and 6 were removed in stages 4 and 5 respectively. Inspite of their removal the packet loss stayed approximately the same in these stages and did not increase. This is because both nodes 5 and 6 were not participating in the flow in the stage prior to the one in which they were removed. Since they were not a part of the flow, removing them did not change the flow capacity significantly to incur increase in the packet loss. This effect is more pronouned in stage 5 where the packet loss stayed the same even though the connectivity plummeted down from 0.8 to 0.08.



(a)

(b)

(c)

(d)

(e)

Figure 15: Node failure results

6.5 Mobility

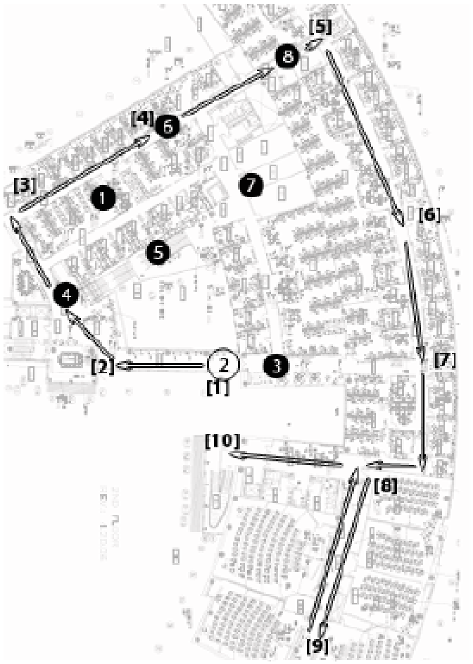
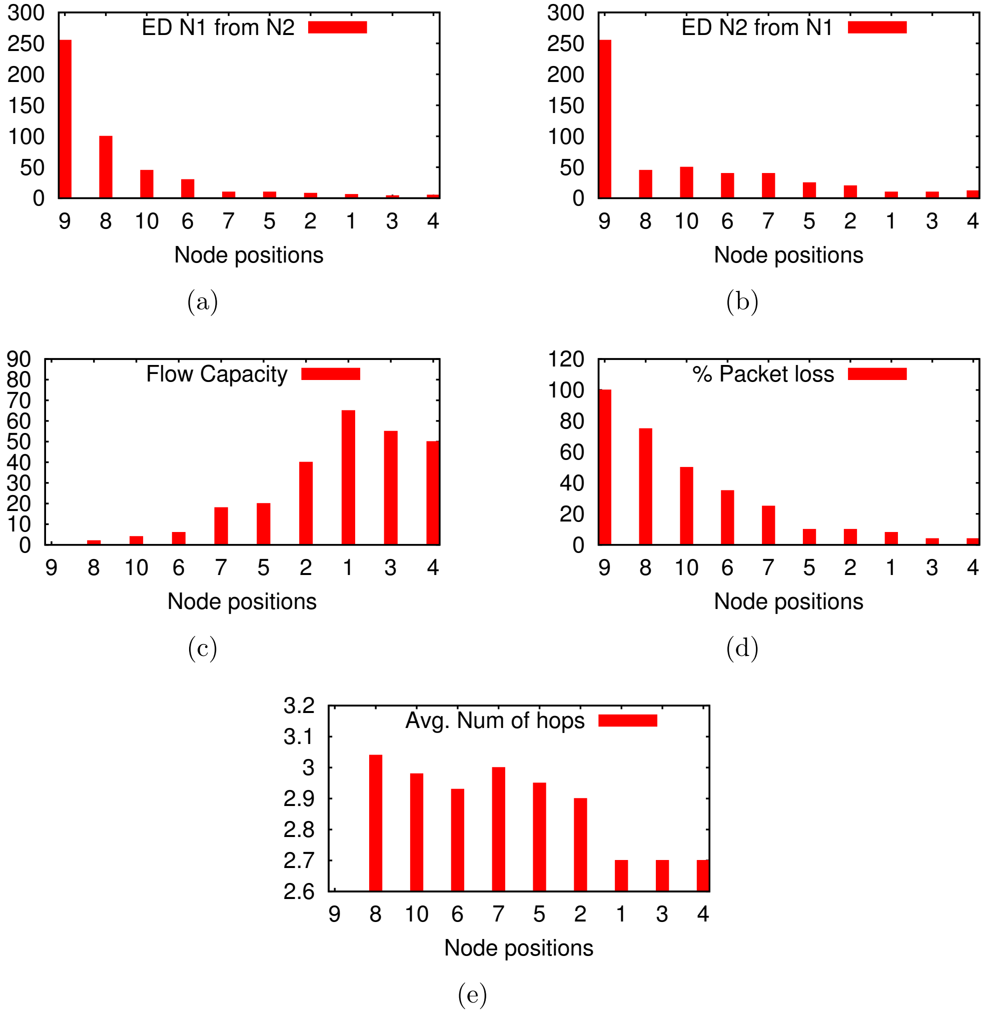


Figure 16: Mobility experiment setup

The experiment that was conducted to see how mobility is handled, followed the experimental setup shown in Figure 16. The experimental setup consisted of eight nodes, seven of them were stationary (*nodes 1,3,4,5,6,7,8*) and one node was mobile (*node 2*). The mobile node (i.e. *node 2*) started from position [1] and traversed along the path outlined by the arrows. Readings were noted at the positions indicated in Figure 16. The experiment ended when *node 2* reached position [10].

6.5.1 Experiment 5

Traffic was maintained between *node 1* (destination) and 2 (source) throughout the experiment duration. Traffic consisted of ping packets with interval set to 1 second. Effective distances, end-to-end ping packet loss, flow capacity and the weighted average number of hops (by packets) were measured at each position. Figure 17 summarizes the experimental results. Results in figure 17 are sorted according to decreasing packet loss values. It can be seen from figures 16 and 17 that effective distances were highest when the mobile *node 2* was farthest from *node 1* at positions [9], [8], [10]. Effective distances were lowest when node 2 was closest to *node 1* (positions [3], [4]) and moderate at the remaining positions[1], [2], [5], [6] and [7]. The effects of mobility were accurately reflected in effective distance values (Hypothesis 7).



(a)

(b)

(c)

(d)

(e)

Figure 17: Mobility results

Figure 17(d) shows that the packet loss values exactly followed the effective distance (figures 17(a), 17(b))) trends. Positions farthest from node 1 ([9], [8]) were marked by highest effective distance (lowest reachability), packet loss, number of hops and lowest flow capacity. Whereas, when node 2 was closest to *node 1* ([3], [4]) Effective distances, packet loss and number of hops were lowest (highest reachability) and the flow capacity was highest. At remaining positions, packet loss, flow capacity and number of hops were also found to be consistent with the effective distances. Throughout the traversal of the mobile node, the traffic flow remained smooth except at position [9], where the mobile node went out of the range of the remaining network. But the flow resumed when it backtracked to position 10 and came back into the range of the network again. Effective distance values correctly captured mobility effects and flexible routing used them to appropriately route packets. Since packet loss, flow capacity and the average number of hops were fairly distributed according to the positions of the mobile node, we claim that mobility was appropriately handled by flexible routing (Hypothesis 8).

6.6 Throughput

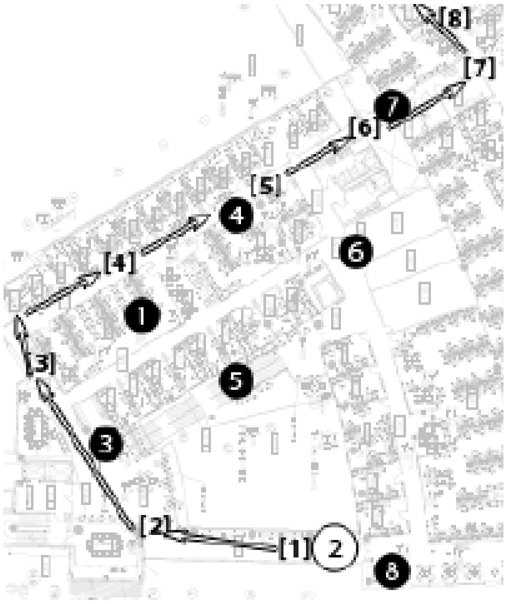


Figure 18: Throughput experiment setup

Although improving throughput was not the goal of our work, we present throughput measurements to give a holistic picture of the system. Figure 18 shows the experimental setup. It consisted of eight nodes, seven of them were stationary (nodes 1,3,4,5,6,7,8) and one node was mobile (node 2). The mobile node (i.e. node 2) began from position [1] and traversed along the path outlined by the arrows. Once it began to move from position [1], it was continuously in motion until position [8], where the experiment ended. Thoughput (TCP) measurements were noted every 10 seconds using the tool Iperf (cite iperf). Figure 19 shows the results.

6.7 Controlling redundancy

Redundancy is necessary for sparse networks. However, it can be an overhead in dense networks or networks containing dense clusters of nodes. Theoretically, the rise in packets would be bounded by $O(n^{TTL})$ as the network scales. But the observed rise in packets would be close to the upper bound only in cases of dense network zones. Dense network zones would be marked by very high connectivity values. Thus, in dense zones redundancy should be controlled for both EDM and data packets.

Unnecessary redundancy in a network can be controlled if unnecessary packet forwarding is restricted in dense zones. When a node in any dense cluster receives packet, chances are high that all other nodes in the cluster have received the packet as well. Hence, it would suffice if only a subset of nodes in the dense zone forward the packet. This is achieved by implementing a simple probabilistic rule for forwarding packets. Every node forwards packets with a probability, which is kept high in sparse networks and low in dense networks. The formula:

$$P_{fwd} = \begin{cases} 1 & \text{if } K <= 3 \\ \frac{3}{K} & otherwise \end{cases}$$

where K represents the number of nodes, whose effective distances are less than ED_{TH} from the node making the forwarding decision. ED_{TH} represents the threshold effective distance; nodes within which are considered close to the node making the forwarding decision. For maintaining consistency, this threshold effective distance was kept similar to the threshol defined for connectivity.

We did not design experiments specifically to evaluated this feature. However, this feature was enabled throughout the duration of the evaluation process. A marked improvement of performance was observed after implementing this feature (approximately 30 percent improvement in packet loss in dense network setups of upto 8 nodes).

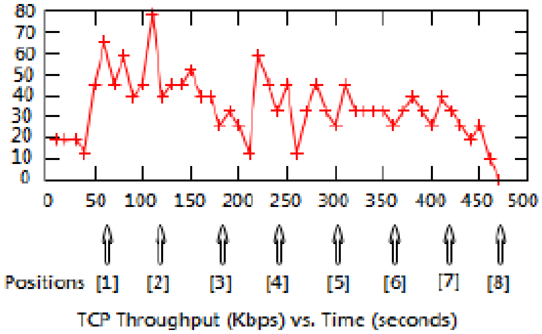


Figure 19: Throughput Results

CHAPTER VII

FIELD TESTS AND DEPLOYMENT PLANS

The design of LifeNet has been an interative process. Our focus has always been on building a system that satisfies critical needs as against trying to fit an already existing solution into some existing problem. Users were hence involved from the early stages of system design.

7.1 First field test with FAA

After the first fully functional proof of concept implementation of LifeNet was complete in Dec 2009, we began contacting organizations and agencies that might have any interest in using LifeNet. The Federal Aviation Authority (FAA) expressed interest in evaluating LifeNet and providing inputs to its further development. FAA engages itself heavily in disaster relief activities and hence agreed to participate in the evaluation of LifeNet. The outdoor field test that was conducted at a third party location primarily aimed at demonstrating the key features of LifeNet to FAA representatives. Mr. Alan Stensland represented FAA during the field test. The test was conducted using 5 nodes, one of which was Mr. Stensland's laptop. Mr. Stensland used standard disaster relief MIS softwares on a LifeNet network to try out several use cases such as creating incident report, logging on to FAA servers using the LifeNet gateway, submitting damage assessment reports, checking email, etc. Overall, he could carry out all the use-cases to his satisfaction. Moreover, he supported our argument of using ad hoc wireless connectivity for operations such as disaster relief and encouraged us to develop and refine the LifeNet prototype further.

7.2 Porting LifeNet on OpenWRT and Android

The first fully functional prototype of LifeNet was implemented using Linux. Naturally, it could be installed and run on most Linux based laptops. Thorough evaluation was done specifically for the Ubuntu distribution. After our first drill with FAA, it was clear to us that to us that the laptops as such had limited communication range and WiFi routers, if used could boost the range significantly. Fortunately, OpenWRT came to our rescue. OpenWRT is a highly extensible GNU/Linux distribution for embedded devices. Since LifeNet code was written for Linux, porting it onto OpenWRT was pretty straight forward. It was an important breakthrough because it reinforced our argument of interoperable design and brought in more flexiblity into the system.

Immediately after our first few field visits (see following sections), we realized that it would be very difficult for all users to carry laptops once they are on field. We needed LifeNet to work on end user devices that are more compact and portable than laptops. Smartphones seemed a promising alternative. Since LifeNet software had both user-level and kernel level modules, we needed a smartphone that allowed us superuser privileges and the ability to load kernel modules. The Android platform was the only smartphone platform, which offered us this flexibility. In comparison with OpenWRT, porting of LifeNet on Android was a considerably complicated effort. This is because, the procedure of porting any Linux-based software that involves a lot of native code (both user and kernel level) is not clearly documented anywhere. The porting effort could be divided in 4 phases as follows:

1. Preparation phase, which involved downloading and installing the software that was necessary for the port. It also consisted of reading hundreds of web-pages, gleaning useful information from them and obtaining a clear understanding of every step involved.

2. Cross-compilation of the user-level native code.

3. Cross-compilation of the kernel-level native code.

4. Packaging as an Android app

For a detailed account on how LifeNet was ported on Android, please read [31].

7.3 Collaboration with Tata Institute of Social Sciences, India

Using inputs from our first outdoor drill with FAA, we refined LifeNet further, ported it onto OpenWRT and Android, built messaging and network visualization applications that would be useful during disaster relief operations. Subsequently we presented a poster in the International Humanitarian Logistics Conference in March 09. It was during the conference that we met Dr. Janki Andharia, the Director of Jamshetji Tata Centre for Disaster Management (JTCDM), Tata Institute of Social Sciences (TISS), Mumbai, India. She expressed interest in a collaboration between Georgia Tech and TISS for the possibility of deploying LifeNet on field. JTCDM is a part of TISS, India, which offers graduate courses in disaster management. In Summer 10, I gave a talk at TISS in which, I presented and demonstrated the key ideas behind LifeNet. A formal collaboration was then formed between Georgia Tech and TISS in Fall 10, with the intent of getting LifeNet deployed on field. It was agreed upon that Georgia Tech would handle the technology aspects whereas TISS would play a key role in on-field deployment operations.

7.4 Field Visits

7.4.1 Kick-off meeting with Maharashtra State Relief and Rehabilitation Cell officials

As a first step of the deployment of LifeNet, a meeting was held with the Secretary of Relief and Rehabilitation Cell of the State of Maharashtra, where LifeNet was presented. The government officials present at the meeting liked the technology behind

LifeNet however they suggested us to first prove the on-field utility of LifeNet by conducting a pilot successfully before involving the government. With help from TISS, we began shortlisting prospective areas in India for conducting a pilot deployment. We began hunting for locations near Mumbai first for logistical flexibility.

7.4.2 Selection of Guhagar and initial findings

After some field visits, we finalized a cyclone prone location called Guhagar, not far from Mumbai and suitable for deploying LifeNet. Amit and Soma from TISS conducted the first detailed feasibility survey in Guhagar. Guhagar is situated on the western coast of India between the Sahyadri mountain range and the Arabian Sea. It is surrounded by Konkan, a narrow 720 Km strip running parallel to the coastline. The beach is locked with two hilly regions of height approximately 400 feet. The town of Guhagar and the village Asgoli is situated at an approximate sea level of 40 feet. A cyclonic storm called 'Phyan' crossed the coast of Guhagar on 11 November 2009. Around 44 fishermen went missing during the storm. Along with loss of lives, the cyclone brought about considerable damage to infrastructure. Electricity posts were uprooted, landlines and mobile connectivity was hampered due to loss of power and several roads were damanged hindering the disaster relief activities. After doing the survey we found out that the reasons why Guhagar was badly affected by Phyan were:

1. The villagers were not prepared to handle a cyclone of such a scale.

2. There was a huge communication gap between the Indian Meteorological Department and local government authorities due to which, the warnings about the approaching cyclone and their seriousness were not properly conveyed to the villagers and in particular, fishermen, who suffered the most.

3. There existed and still exists an acute shortage of skilled human resources. There is only one engineer in the local technical staff cadre.

4. There was and still exists heavy reliance on locals for damage assessment. The administration does not have adequate machinery and other equipment needed for search and rescue work.

5. The government mainly relies on mobile communication and landline telephones. Given a condition such as a cyclone, these communication media are often renderred useless. There does not exist any shortage of alternate communication equipment such as VSAT and wireless sets due to very high equipment costs.

7.4.3 Initial lessons learnt

Subsequent field visits to Guhagar were mainly aimed at establishing contacts on the field and establishing relationships. After interacting with local government officials, personnel from local NGOs and disaster-affected victims like fishermen we understood critical points that are necessary to build a sustainable solution.

- **Minimum use of infrastructure** - During Phyan mobile communication was hampered due to failure of existing telecommunication infrastructure. Moreover, repair activities were delayed by weeks due to destruction of roads. This remains the case for all other medium and large scale disasters. Hence, in order to be feasible and sustainable the solution should be capable of efficient operation using of minimum infrastructure.

- **Providing responsibility to local residents** - In developing countries, the government lacks sufficient workforce in disaster situations due to insufficient resources, particularly in remote rural areas. Hence, there exists a greater reliance on local village residents for damage assessment information, incident reporting and actual rescue activities. We learnt that the proposed solution should give due consideration to this fact in order to be sustailable.

- **Minimal power consumption** - All medium and large scale disasters result in loss of power. The cyclone Phyan was no exception. Secondly, restoration of power may take long (few weeks) depending on the exact situation. In the case of Phyan, most of the electricity poles were uprooted and even partial restoration of power took 2 weeks. Hence, consuming minimal power is a critical and indispensible requirement. The communication equipment should operate on batteries and should last sufficiently long to have any impact.

- **Locally maintainable communication equipment** - This lesson is often ignored by people who deploy communication equipment for disaster communication. Guhagar is an apt example. The government of Maharashtra state had distributed 22 Motorola wireless sets to selected fishermen from Guhagar to help them communicate inside the sea when they go fishing. There were two major problems with these sets:

  1. Their cost was very high.
  2. They were not locally maintainable. We found that many wireless sets were rendered unusable due to practical difficulties faced by their owners in sending them to Motorola and getting them repaired.

  Hence we argue that the equipment used should be readily available in local markets and readily maintainable as well.

- **Making the solution a part of users' daily life** - Since it is important to involve local residents in disaster relief activities, convincing them to learn the solution is also extremely important. This is only possible if they have enough motivation to learn the solution. This motivation can only arise if the solution adds a value to their daily life, in the form of some service, which they would be willing to use everyday.

CHAPTER VIII

CONCLUSIONS

The work that began around 3 years ago took concrete shape in August 09 when Ashwin and Santosh came up with the first prototype called MyMANET and subsequently presented its early evaluation in [34]. Immediately after the first prototype had all the minimal required functionalities, we started building relationships with field partners since feedback from real users directed our design in the needed direction. We wanted to be sure that we were solving the right problems.

On the technical side, we focused on understanding the behaviour of the LifeNet design by conducting continuous detailed measurements. In that process, we identified and overcame some major design flaws. For example, MyMANET was based on end-to-end packet loss as a routing metric. During some of our experiments we observed that the routing metric based on packet loss did not capture availability on less utilized paths very well and hence was an inaccurate representation of the network state. This finding then led to the conception of the reachability metric for LifeNet, which was observed to be much more accurate. As shown in Chapter 6, reachability accurately captures all aspects of transience. Consider another example. Results of one of these measurements showed that redundancy can be very high in dense networks or dense network zones. We then implemented probabilistic forwarding rules in the flexible routing algorithm that drastically improved the network performance. Chapter 6 also shows that the flexible routing protocol is capable of reliable packet delivery under varying degrees of transience. In other words, the design of the system has always been driven by the results of these continuous measurements. We have also focused on making LifeNet interoperable with various software and hardware platforms. LifeNet, which was initially designed as a Linux kernel module, is now implemented for the Android and OpenWRT [39] operating systems. This means that along with laptops, LifeNet can also run on any off-the-shelf Wifi router and Android phone. We also have a proof-of-concept implementation of LifeNet over the Microsoft Windows operating system, which would soon be extended with addition of all functionalities.

One of our key realizations is the fact that it is very difficult to achieve even a practical tradeoff between the mutually conflicting goals of high throughput and high reliability under transience (fault-tolerance). Both goals cannot be satisfied at the same time in multihop ad hoc wireless networks. As per our evaluation results, the only promising way to handle transience is to achieve high fault-tolerance by employing a completely distributed multipath routing approach. However, it is not possible to achieve high throughput if multipath routing is used. One of the important reasons for this is the fact that most commonly used wireless MACs are optimized for unicast traffic only. Just to put this argument into perspective, our results show that for a single wireless link between two nodes, throughput degrades ten times when broadcast is used instead of unicast. This is because, broadcasting lacks sufficient help from the MAC layer in the form of MAC level acknowledgements and other optimizations. On the other hand, our results also indicate that for multipath routing, using multiple unicasts instead of broadcasts is also extremely inefficient due to problems such as MAC level buffer overflows. Hence the flexible routing algorithm has to exploit the multi-access nature of the wireless channel by using broadcasts to efficiently routes packets on multiple paths. The price for this approach is then the throughput.

One of the most important challenges that we are yet to fully overcome, is addressing the the power issue as the network scales. We are currently focusing on making the flexible routing protocol more power-aware. But as far as WiFi is concerned, the problem lies more in the MAC layer itself. However, we are optimistic because more and more vendors are now coming out with low power WiFi radios. Moreover, if need be we would also consider implementing LifeNet over alternative communication protocols like ZigBee [14] for low power needs.

Building field relationships has always been one of our important goals. In that effort, we formed a collaboration with the Jamshetji Tata Centre for Disaster Management at Tata Institute of Social Sciences India, which proved very valuable (Fall 2010). Simultaneously as we addressed technical challenges of LifeNet in our research lab, we kept on receiving valuable feedback from the TISS team. With this feedback cycle, we are now confident about LifeNet's design and that it satisfies critical communication needs in disaster situations.

REFERENCES

[1] "Distance-vector routing protocol," *Wikipedia.*

[2] "Link-state routing protocol," *Wikipedia.*

[3] *Split multipath routing with maximally disjoint paths in ad hoc networks*, vol. 10, Aug. 2002.

[4] "Optimized link state routing protocol (olsr)," 2003.

[5] Akkaya, K. and Younis, M., "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, pp. 325–349, 2005.

[6] AL-KARAKI, J. N. and KAMAL, A. E., "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, pp. 6–28, 2004.

[7] Biswas, S. and Morris, R., "Exor: Opportunistic multi-hop routing for wireless networks," in *in SIGCOMM*, pp. 133–144, 2005.

[8] Brewer, E., "A certain freedom: thoughts on the cap theorem," in *Proceeding of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, PODC '10, (New York, NY, USA), pp. 335–335, ACM, 2010.

[9] Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., and Jetcheva, J., "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '98, (New York, NY, USA), pp. 85–97, ACM, 1998.

[10] Dahlman, E., Parkvall, S., Skold, J., and Beming, P., *3G Evolution, Second Edition: HSPA and LTE for Mobile Broadband.* Academic Press, 2 ed., 2008.

[11] Das, S. R. and Perkins, C. E., "Performance comparison of two on-demand routing protocols for ad hoc networks," pp. 3–12, 2000.

[12] De Couto, D. S. J., Aguayo, D., Bicket, J., and Morris, R., "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03)*, (San Diego, California), September 2003.

[13] Draves, R., Padhye, J., and Zill, B., "Routing in multi-radio, multi-hop wireless mesh networks," in *In ACM MobiCom*, pp. 114–128, ACM Press, 2004.

[14] Farahani, S., *ZigBee Wireless Networks and Transceivers*. Newton, MA, USA: Newnes, 2008.

[15] Fox, A., Gribble, S. D., Chawathe, Y., Brewer, E. A., and Gauthier, P., "Cluster-based scalable network services," *SIGOPS Oper. Syst. Rev.*, vol. 31, pp. 78–91, October 1997.

[16] Gilbert, S. and Lynch, N., "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *SIGACT News*, vol. 33, pp. 51–59, June 2002.

[17] Grossglauser, M. and Tse, D., "Mobility increases the capacity of ad-hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 477–486, 2002.

[18] Gupta, P. and Kumar, P. R., "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, March 2000.

[19] Haas, Z. J. and Pearlman, M. R., "The zone routing protocol: A hybrid framework for routing in ad hoc networks," *Ad Hoc Networks*, 2000.

[20] Härder, T. and Reuter, A., "Principles of transaction-oriented database recovery," *ACM Comput. Surv.*, vol. 15, no. 4, pp. 287–317, 1983.

[21] Harras, K. A. and Almeroth, K. C., "Controlled flooding in disconnected sparse mobile networks," *Wirel. Commun. Mob. Comput.*, vol. 9, pp. 21–33, January 2009.

[22] Hedrick, C., "An introduction to igrp," 1989.

[23] Intanagonwiwat, C., Govindan, R., and Estrin, D., "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, (New York, NY, USA), pp. 56–67, ACM, 2000.

[24] Jain, K., Padhye, J., Padmanabhan, V. N., and Qiu, L., "Impact of interference on multi-hop wireless network performance," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, MobiCom '03, (New York, NY, USA), pp. 66–80, ACM, 2003.

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

[26] Johnson, D. B., Maltz, D. A., and Hu, Y. C., "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," tech. rep., IETF MANET Working Group, Feb. 2007.

[27] Kanade, V. and Vempala, S., "Life (and routing) on the Wireless Manifold," 2007.

[28] Karger, D. R. and Tai, R. P., "Implementing a fully polynomial time approximation scheme for all terminal network reliability," in *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, SODA '97, (Philadelphia, PA, USA), pp. 334–343, Society for Industrial and Applied Mathematics, 1997.

[29] Li, J., Blake, C., De Couto, D. S. J., Lee, H. I., and Morris, R., "Capacity of Ad Hoc wireless networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, (New York, NY, USA), pp. 61–69, ACM, 2001.

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

[31] Mehendale, H., "Lifenet blog: Porting lifenet on android (http://thelifenetwork.org/blog/index.php/2011/05/28/porting-lifenet-on-android-cyanogenmod)," 2011.

[32] Mouly, M. and Pautet, M.-B., *The GSM System for Mobile Communications.* Telecom Publishing, 1992.

[33] Moy, J., "Ospf version 2," 1998.

[34] Paranjpe, A. and Vempala, S., "Mymanet: A customizable mobile ad hoc network," in *NSDR '09*, (Big Sky, Montana, USA), ACM, October 2009.

[35] Park, V. and Corson, M. S., "Tora : Temporally ordered routing algorithm," in *IEEE INFOCOM.*

[36] Pei, D., Massey, D., and Zhang, L., "A formal specification for rip protocol,"

[37] Perkins, C., Royer, E., and Das, S., "RFC 3561 Ad hoc On-Demand Distance Vector (AODV) Routing," tech. rep., 2003.

[38] Perkins, C. E. and Bhagwat, P., "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, 1994.

[39] Petullo, M., "Building custom firmware with openwrt," *Linux J.*, vol. 2010, August 2010.

[40] Press, A., "Number of cell phones worldwide hits 4.6b," *CBS News*, February 2010.

[41] Raman, B. and Chebrolu, K., "Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks," in *Proceedings of the 11th annual international conference on Mobile computing and networking*, MobiCom '05, (New York, NY, USA), pp. 156–169, ACM, 2005.

[42] Vahdat, A. and Becker, D., "Epidemic routing for partially-connected ad hoc networks," tech. rep., 2000.

[43] Viterbi, A. J., "Cdma: principles of spread spectrum communication," *IEEE Wireless Communications*, 1995.

[44] Walke, B., Seidenberg, P., and Althoff, M. P., *UMTS: The Fundamentals.* Wiley, 2003.

[45] Woo, A., Tong, T., and Culler, D., "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, (New York, NY, USA), pp. 14–27, ACM, 2003.

[46] wu Chin, K., Judge, J., Williams, A., and Kermode, R., "Implementation experience with manet routing protocols," *ACM SIGCOMM Computer Communications Review*, vol. 32, pp. 49–59, 2002.

[47] Yarvis, M. D., Conner, W. S., Krishnamurthy, L., Mainwaring, A., Chhabra, J., and Elliott, B., "Real-world experiences with an interactive ad hoc sensor network," 2002.

[48] Zhao, W., Ammar, M., and Zegura, E., "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '04, (New York, NY, USA), pp. 187–198, ACM, 2004.