Opportunistic Communication Schemes for Unmanned Vehicles in Urban Search and Rescue

by

Sion Scone

A Doctoral Thesis

Submitted in partial fulfilment
of the requirements for the award of

Doctor of Philosophy
of
Loughborough University

29th June 2010

Loughborough University

Thesis Access Form

Copy No……..……**Location**…….……..…

Author……..……..…….

Title……..

Status of access OPEN / RESTRICTED / CONFIDENTIAL

Moratorium Period:……years,ending……../……200…….

Conditions of access approved by (CAPITALS):……

Director of Research (Signature)……...……...

Department of……...……

Author's Declaration: *I agree the following conditions:*

OPEN access work shall be made available (in the University and externally) and reproduced as necessary at the discretion of the University Librarian or Head of Department. It may also be copied by the British Library in microfilm or other form for supply to requesting libraries or individuals, subject to an indication of intended use for non-publishing purposes in the following form, placed on the copy and on any covering document or label.
*The statement itself shall apply to **ALL** copies:*

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Restricted/confidential work: All access and any photocopying shall be strictly subject to written permission from the University Head of Department and any external sponsor, if any.

Author's signature…….Date……...……...……...

users declaration: for signature during any Moratorium period (Not Open work): ***I undertake to uphold the above conditions:***			
Date	Name (CAPITALS)	Signature	Address

Research Student Office, Academic Registry
Loughborough University, Leicestershire, LE11 3TU, UK
Switchboard: +44 (0)1509 263171 Fax: +44 (0)1509 223938

Loughborough University

Certificate of Originality

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgements or in footnotes, and that neither the thesis nor the original work contained therein has been submitted to this or any other institution for a higher degree.

. . . . . .

Sion Scone

-29th June 2010

Acknowledgements

I would like to thank:

- Dr Iain Phillips for his guidance, supervision and support throughout the course of my PhD.
- Dr Jim Armstrong for his expert advice during the early phases of my research.
- Brian Ford for his ever-helpful guidance while learning to use the Koala robots in the Consert rig.
- Jordan Labrom for his help with the statistical analysis of the MVT function, particularly in generating the gain curves and tangents.
- Dr Qinggang Meng for reading my work and providing feedback.
- Everyone in the Network Research Group for listening to my work and critiquing.

Opportunistic Communication Schemes for Unmanned Vehicles in Urban Search and Rescue

Sion Scone

June 29, 2010

1 Abstract

In urban search and rescue (USAR) operations, there is a considerable amount of danger faced by rescuers. The use of mobile robots can alleviate this issue. Coordinating the search effort is made more difficult by the communication issues typically faced in these environments, such that communication is often restricted. With small numbers of robots, it is necessary to break communication links in order to explore the entire environment. The robots can be viewed as a broken ad hoc network, relying on opportunistic contact in order to share data. In order to minimise overheads when exchanging data, a novel algorithm for data exchange has been created which maintains the propagation speed of flooding while reducing overheads. Since the rescue workers outside of the structure need to know the location of any victims, the task of finding their locations is two parted: 1) to locate the victims (Search Time), and 2) to get this data outside the structure (Delay Time). Communication with the outside is assumed to be performed by a static robot designated as the Command Station. Since it is unlikely that there will be sufficient robots to provide full communications coverage of the area, robots that discover victims are faced with the difficult decision of whether they should continue searching or return with the victim data. We investigate a variety of search techniques and see how the application of biological foraging models can help to streamline the search process, while we have also implemented an opportunistic network to ensure that data are shared whenever robots come within line of sight of each other or the Command Station. We examine this trade-off between performing a search and communicating the results.

2 Keywords

Urban Search and Rescue, Mobile Robots, Communication, Opportunistic Networks, Mobile Ad Hoc Networks, Marginal Value Theorem, Search Strategies

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Urban search and rescue (USAR) teams operate in extremely hazardous environments. Area reconnaissance is required prior to committing rescue workers to the search area; each hazard needs to be fully assessed prior to entry, and as such, deploying rescue workers can be a slow process. Qualified rescue workers are almost always in short supply. There is a direct correlation between the amount of time taken to extract victims and their survival rates: the faster they are rescued, the more likely they are to survive. Therefore, reducing the rescue worker bottleneck could lead to increased survival rates. Mobile unmanned autonomous or semi-autonomous vehicles or robots could be deployed to help speed up area reconnaissance or search.

The first deployment of mobile robots in USAR occurred at the World Trade Centre disaster in September 2001. Often the limiting factor of the robot's capabilities are sensors, and so teams of inexpensive robots are generally preferred to a single robot in order to provide multiple readings. Communicating between these teams can lead to increased effectiveness, but USAR environments are inherently noisy and communication is often limited to line-of-sight. By treating each robot as a node in a network, the USAR team can be viewed as a sparsely connected ad hoc network where network joins and breaks occur frequently, limiting the use of traditional routing schemes. It is often necessary to have a guaranteed quality of service link between two points - for instance, to provide an end-to-end video link from rescuers to an entombed victim - but often nodes will be forced to operate outside of communication range of any base station (where data can be collated for analysis), meaning that they are effectively cut off. Whenever any node comes back within range of the base station, whatever data it has collected can be exchanged and used to update the rescuer's world model. However, this requires that each node periodically moves within communication range of another in order to pass data on; potentially this might involve halting the search in order to

communicate findings. It is important to note that while our discussions assume the deployment of mobile robots, the same principles apply to wearable computer systems such as those worn by firefighters to monitor local conditions inside burning buildings. Therefore the term 'node' could apply to human rescue workers as well as mobile robots. A full background survey can be seen in Chapter 2.

In Chapter 3 there is a full description of how the experimental environments were set up, and how the necessary behaviours and communication systems were implemented, as well as a description of the experiment methodology.

In order to reduce the need to report data back to a base station, each node can make use of opportunistic contacts with other nodes in order to exchange all onboard data regardless of origin. Making use of these opportunistic contacts to exchange data ensures that more data reaches the base station - or 'Command Station' - more quickly without imposing any movement restrictions on the nodes, ensuring that their primary task is unaffected. The effectiveness of opportunistic communication is examined via a search task in Chapter 4.

Since wireless communication costs energy, and all mobile robots or wearable computers must operate on battery power, reducing the amount of communication required, and will result in saved energy. This relates to an increase in battery life, allowing robots to remain in the field for longer, and therefore to provide a better service. We examine how to streamline data propagation with an efficient data exchange mechanism for mobile nodes in an opportunistic network in Chapter 5, while ensuring that latency is minimised.

During any search operation where nodes are not within communication range of the Command Station and there is no global communication, there will be a tradeoff between the time taken to search the environment, and the delay to get the data back to the Command Station. If a node continues to search after a victim is discovered then the overall search time will be minimised but the delay in getting that information back to the base station will be increased. If a node immediately returns to within communication range of the base station to report the location of the victim before continuing the search, then the delay time will be minimised but the search time will be increased. In Chapter 6 we examine this tradeoff by comparison of various strategies with a view to minimising the total search time, and conclude findings and future work in Chapter 7.

Chapter 2

Communication Issues in USAR

2.1 Background

Urban Search and Rescue (USAR) deals with the location and extraction of victims in or around collapsed buildings and other structures in the aftermath of natural disasters such as earthquakes, or in cases of war, kidnap or terrorism. In the UK, USAR falls under the responsibility of the Fire Service [24]. Other search and rescue operations deal with searches in other environments, most often mountains or sea. This kind of role is often served by the military responding to emergencies such as the floods or hurricanes [68, 42].

The use of unmanned vehicles in non-urban search and rescue operations is limited mainly to locating victims and detecting damage from the air [42], tasks that cannot typically be performed in USAR operations. This thesis will focus on the communication issues when using unmanned vehicles in USAR.

2.1.1 Characteristics of USAR Sites

The physical characteristics of an USAR site vary widely. Buildings are often collapsed or have lost structural integrity and have the potential for further collapse [57]. Partial building collapse exposes internal components of the building that are not designed for exposure to rain to absorb water, increasing the possibility of further collapse; high winds and after shocks can also cause collapse, which can create further casualties [10, 14]. The immediate area around the collapse or explosion tends to be chaotic, with piles of rubble and sharp objects protuding in all directions [10]. There may be a host of hazardous materials, lack of oxygen due to poor ventilation or a build up of gases, electrical interference, fires, standing water or sewerage, sharp objects which bring a risk of entrapment, toxic contamination, or unstable elements such as explosives [10, 57]. Building characteristics can vary widely dependent upon local building materials and culture, which can

affect both the types of collapse that are common and also the kinds of hazards that will be prevalent [10]. Weather conditions vary widely; rain, snow, sleet, wind, cold and changes in light can all cause conditions to change rapidly. The aftermath of the World Trade Centre collapse left large rubble piles with small voids that required penetration, while Hurricane Katrina resulted in a flooded area with thick mud [52]. It is therefore difficult to characterise a 'typical' USAR environment.

2.1.2 Organisation of Response

USAR operations are generally large scale and require specialists to gather from geographically distant locations. US government legislation dictates that only trained rescue workers may enter a rescue site [57]; outside the US, experienced rescue workers aim to quickly assess the site and prevent access to non-qualified personnel. This means the number of rescue workers within the site is always likely to be low. It is vital that all USAR operations have a common response method in order to minimise the time required to organise the rescue efforts [14]. This includes the creation of a 'Command Station' from which the operation can be coordinated and managed [41]. Because each USAR incident is unique, management tools need to be flexible and scalable [14]. The Command Station can then send data to other rescue workers in a geographically distant location where it can be analysed. This allows other qualified and experienced rescue workers to help even if they cannot be at the disaster site, and is referred to as 'reachback'. Reachback is designed to allow access to information and resources regardless of location in order to maximise the number of trained personnel who can assist in the rescue. The current practice is to use phone or fax to communicate between local and remote groups, but this adds additional disruption as data must be validated by experts [32]. A recently developed system by Rehmani et al [71] created a framework which uses cognitive radio (CR) technology to provide connectivity to damaged or partially destroyed networks. CR devices can perform discovery of existing, yet damaged, infrastructure and devices, then tune in and act as a data relay to restore connectivity.

2.1.3 Area Assessment

Firstly the area around the site must be assessed, and divided into three zones: a 'hot' zone representing the rescue site itself, where only people and equipment essential to the rescue effort are allowed; a 'warm' zone which is the location of the Command Station, victims families and equipment storage; and a 'cold' zone which is anywhere outside these two areas and is open to the public [57]. It is

essential that the hot zone is assessed for hazards, and that entrance to the hot zone is prohibited to all rescue workers until the area assessment is completed [14]; even rescue dogs cannot enter some voids, and are just as vulnerable to hazards as humans [10]. There are often a great deal of people around the rescue site, including the victim's families, good Samaritans and the media, all of which complicate management of the rescue [10]. High ambient noise levels due to power generators, chain saws and other heavy machinery, press helicopters and constant radio traffic can make communication difficult. Inside the structure, noise levels drop dramatically and it is possible to hear cries for help; distinguishing between noise and genuine cries for help can be made easier by reducing the noise level outside [10, 57]. Efforts should be concentrated on finding and releasing survivors, yet distinguishing between survivors, dead bodies, and false readings due to distractions that can be thought to be survivors, such as clothing or CO_2 emissions is problematic [10]. Because of these factors and government legislation, only trained and certified rescue workers may enter the hot zone [57]. The hot zone itself is generally difficult to explore, and the existence of hazards force rescuers to work very slowly. Rescue efforts can be restricted because of the need to stick to safety standards [10, 57]. In addition, rescue workers suffer sensory deprivation from their own safety equipment such as masks and gloves which can further slow down rescue attempts [10]. In fires, sensory deprivation makes it difficult to assess casualties inside, so all casualties are generally removed; this can result in dead bodies being pulled out while others are still alive [31].

Each area beneath a collapsed structure is called a void, and the characteristics of the void can be recorded and assessed by structural engineers so that rescuers can plan an appropriate response [14, 54]. These characteristics include, but are not limited to, the internal volume of the void, minimum cross section through which a rescue worker might enter, size and material type of debris, wetness, grade of flooring, temperature, variability of these conditions, and whether there are any hazards such as chemical or electrical hazards [54]. This can then form the basis of a coordinated rescue attempt. It is not only the victims that are at risk during a rescue attempt; USAR sites are dangerous to rescuers as well. Therefore it is vital that collateral threats such as those mentioned in Section 2.1.1 are assessed and removed before rescue workers can access the site [10, 14]. This increases the time that a successful rescue takes, but is vital; in the 1985 Mexico City earthquake, 65 rescue workers drowned when the void they were working in was flooded [14]. Once hazards have been dealt with, voids can be searched and victims and debris can be removed with the assistance of structural engineers [14].

2.1.4 Victim Welfare

Typically, rescue efforts take a minimum of four hours and can carry on around the clock for several days [57]. The chances of surviving entrapment decrease rapidly with time; if rescued within 24 hours, 81% survive, while just 7.4% survive after being trapped for five days [14]. The combination of the effort duration, a bottlenecked supply of rescue workers and the time critical nature of the rescue means that those rescuers will soon be operating under stressful and fatiguing conditions [57]. When victims are located, they can be suffering from a variety of medical issues such as hypothermia, crushing of limbs, blood loss, inhalation injuries from dust and fibreglass, broken bones or dislocated joints [14].

2.1.5 Victim Location and Extraction

The first step in victim extraction is of so called 'surface victims' - those hit with debris or injured during a fall - who account for approximately half of all recovered victims. They have the best chance of surviving because they are easily found, reached and released, and their rescue is the most utilitarian action [14]. Despite the fact that these victims might be easily visible and even calling for help, it is vital that the area is not approached until area assessment is complete [14]. The second stage is extraction of 'lightly trapped' victims - those that can be freed by one or two rescuers - accounting for approximately 30% of recovered victims. These victims are currently located by search dogs, listening devices, fibre optic cameras and thermal imaging [14]. Approximately 15% of victims are recovered from voids with an approximate rescue time of four hours, each requiring ten dedicated rescue workers, and the remaining 5% are classed as 'entombed' which means they are trapped by main walls and have an average rescue time of ten hours [14]. Casper et al [14] describe how survival rates fall rapidly in USAR; after 24 hours, 81% of victims survive, compared to just 37% after 48 hours. Backstrom and Christofferson [20] provide further statistics; 39% of all rescue teams arrive in the period 12-24 hrs after the incident, and approximately 70% of all victims are recovered within 12 hrs [20]. Given the information above, we can say that the window of 12-24 hours post-incident is where the majority of the work for USAR teams takes place.

2.1.6 Rescue Operations involving the Fire Services

Fire crews will usually only enter a burning building if they have definite knowledge regarding the presence of casualties [22]. Entry is also dangerous due to the risk of flashovers - with temperature of approximately 1000°Celsius - or backdrafts - with

temperatures of between 460°-800°Celsius - which are far above the 160°Celsius that the firefighter's suits can deal with [31]. Inside, there is virtually zero visibility, and fire fighters are forced to use their hands and feet to explore the space by touch. Smoke nullifies noise, making communication difficult. Generally, by the time fire crews arrive, any victims still inside the building will be unconscious, making it unlikely that victims will respond to audio calls [22]. Building search is conducted by wall-following, starting in the area closest to the fire [31]. Searching is conducted in buddy pairs who remain connected at all times, either by holding hands or via a 6m line; one hugs the wall to the left while the other is to their right towards the middle of the room. Moving in this way the firemen are able to search a 2m wide space, aiming to form a mental picture of the room; however, the experience is extremely disorientating, and it is possible that a middle section of a large room is left unsearched [22]. Firemen report that their mental picture of a room during a fire contrasts hugely with the room viewed after the event [22]. Once a room is searched, doors are closed; closed doors indicate searched spaces.

Firefighters communicate via radio, but frequent loss of communication occurs. The officer in charge is based outside the building with a Merlin control board [29] which shows readouts from the fire fighter's suit, such as temperature, oxygen levels, and estimated air time remaining. If a firefighter's signal is lost on the Merlin board then the officer in charge will attempt to contact them via radio. If this is unsuccessful then the officer attempts to contact the firefighter's buddy. Generally this is successful - a clear indicator that intelligent routing could be incorporated into the existing control board system - but if it is not then this buddy pair will be treated as potential casualties and another crew will need to be sent in after them. In addition, each firefighter's suit is fitted with an Automatic Distress Signal Unit (ADSU) that gives off a warning signal if no movement is detected from the firefighter for a certain amount of time [22].

There is no formal decision making process with regards to committing rescue workers, either to fires or technical rescue; instead, the emergency services rely on the knowledge and experience of their members to perform a risk assessment [27, 31]. A risk assessment can take into account the type of building or structure, physical distance that the casualties are likely to be from the fire, time from start of fire until emergency services were called, response time of firefighters, type of carpets, number of videos/dvds (which burn like petrol), presence of cooking oils (particularly restaurants) or engine oils (garages), and possibility of damage to utilities such as gas, electricity or water. Open doors or windows indicate lower possibility of flashover or backdraft; likewise pulsating smoke indicates high

likelihood of these. Children are more likely to survive than adults, so if there are reports of children inside a building then it is more likely that firefighters will commit to entry. Ultimately, the decision is one of risk to firefighter against chance of success [31].

2.2 Role of Robots in USAR

Unmanned, autonomous robots (UAR) are machines capable of performing various tasks without relying on infrastructure services, remote control or other external inputs [50]. UARs can be deployed for search and rescue, surveillance, or other remote information gathering exercises in areas where humans cannot go themselves to due to physical size (such as after building collapse), physical endangerment (such as flooded mines or toxic gas), or due to risk of capture (military scenarios). UARs come in many shapes and sizes,and can be air, sea or land based (or a combination of those three) [32]. UARs are often used as remote groups, reporting back to a local command centre [32]. Using several UARs introduces redundancy, increasing the chances that the task will be completed and enabling more data to be collected [13]. Many missions where autonomous systems are deployed are dangerous; a single UAR might not survive long enough to complete the task. A UAR deployed in USAR operations needs to be physically small, highly mobile, untethered, with sufficient runtime and high enough speed to complete its mission without recharging, and sufficiently small that they can be carried and deployed by a single person [14, 51, 54, 57]. Each mission will involve tradeoffs; for instance, if speed and mobility is most vital, then the number of sensors may need to be reduced [14, 87].

USAR operations are long, tiring and stressful, and difficult decisions must be made to put rescue workers at harm's way in order to get information about the environment. Many tasks at a USAR operation are repetitive and can easily be automated; UARs could be deployed to ease the load on rescue workers [10, 14]. In USAR incidents, rescue workers often have to go into areas about which there is no prior knowledge. Controlling a robot remotely via teleoperation is the method preferred by rescue workers but has the need for a reliable and continuous remote communication link, which limits range [41]. UARs are deployed into USAR environments in a variety of ways, including ground entry [57], from the air or lowered by tethers [87], or via holes cut in the structure by rescue workers [57]. Technical rescue teams are currently able to penetrate around 18ft into a void that is unenterable by humans, which provides a good benchmark for testing [14]. Ideally, a team of UARs could be deployed at a USAR site and perform a full reconnaissance of the

area and extract all victims without placing a single human rescuer at risk. However, this is beyond the means of current technology, and research is divided into solving individual components of the puzzle such as traversal of unpredictable and unstable terrain [51]; self localisation [41]; robot localisation and navigation using echolocation [88]; production of accurate human-readable maps [41]; a method of passing data between local and remote rescuers and UARs [57]; bespoke design of UAR and sensors [14]; autonomy in case of communications breakdown [65]; cooperation between UARs [65]; and HCI for non-technical users [10]. Hardware errors are also frequent, and therefore redundant equipment should be available in the field where possible [32]; sensors are frequently the weakest link in the entire system [14]. It is therefore feasible to deploy a team of simple, inexpensive, expendable UARs rather than a single, expensive one [73]. Essentially, results from any one UAR should be regarded as potentially unreliable, technology can be expected to fail in stressful situations, and the system should be able to compensate accordingly [32]. Prior to use in a real rescue, new technologies must be proven robust enough to be an aid to the USAR team [52]. Currently there are no baseline tests [52]; until now, systems have been evaluated by mission success in terms of time spent and percentage of the task completed, such as in the RoboCup Rescue Competitions [41, 73].

2.2.1 Potential Tasks for UARs

Once UARs have the necessary functionality to operate inside the hot zone, there are many tasks that they can perform, including the following [41, 57]:

- identification, categorisation and monitoring of potential hazards
- identification of victims regardless of whether they are trapped, covered by rubble or dust, or in different positions
- medical checking of victims; search efforts should not be wasted on the dead while there are survivors still trapped
- act as a communications relay between the Command Station and victims or other UARs that are out of communications range
- assess the stability of partially collapsed structures and provide shoring
- produce accurate maps

We will focus on four of these tasks: Reconnaissance, Search, Communication Relay, and Environmental Monitoring.

2.2.2 Reconnaissance

The first task that must be completed in an USAR operation is a thorough reconnaissance or area assessment to discover the nature of the search site and to establish whether it is safe for rescuers to begin their search [57]. UARs can assist in this task by finding and noting the locations of hazards, of which there are seven major types: 'structural instability', such as weakened floors which people could fall through or walls that could collapse; 'overhead', such as dangling wires or a loose ceiling section which can cause entrapment; 'surface', such as glass, nails or other debris which can cause cutting of skin and equipment; 'below-grad', such as a broken gas pipe in a basement creating an area without oxygen; 'utilities', such as a gas supply that could explode; 'hazardous materials', such as chemicals that can cause corrosion of equipment and safety suits; or 'incident', such as fire or smoke [14]. However, sensors are highly sensitive to noise which can make readings unreliable [13]. Since the location and density of hazards is unknown, it is preferable that UARs cover as much area as possible during their sweep; if any of the identified hazards require continuous monitoring then they can be flagged as a 'region of interest' [4]. These could be out of communications range of the Command Station; hence the need for communications relay [67].

Failure to correctly assess the situation can lead to rescuers being placed in unnecessary danger [14, 57]. Automating reconnaissance can also speed up the rescue as the reconnaissance can be completed during the organisation and creation of the Command Station; the reconnaissance might even find some victims [14]. Once hazards have been located they can be placed onto a human-readable map and continually monitored, and the rescue can be planned with the hazard in mind [41, 57].

2.2.3 Search

The most obvious use of UARs is for searching for victims, often in areas that would not permit human access [57]. There are five signs of human life; human form, body heat, sound, motion and CO_2 emissions, and with the use of sensors UARs can ensure high fidelity results by requiring at least two of these signs before being sure that a victim has been found; false readings are penalised as they could put rescue workers at unneccesary risk [41]. Finding the victim is only the first stage in their recovery; the victim's location needs to be communicated back to the Command Station, which means that the UAR needs very accurate localisation based not only on internal sensors but also relative to its surroundings so that they can be accurately placed on maps [41]. If there is a communications link

back to the Command Station, it is advantageous for the UAR to stay with the victim in order to maintain a communications link via microphone, and to provide continuous monitoring of the victim's health. If contact is established with the victim then they might be able to provide rescuers with additional information such as the presence of other victims or hazards [57]. Autonomous extraction of real victims is deemed incapable of being automated so the role of UARs is currently limited to locating victims [43, 57].

2.2.4 Communication Relay

USAR sites are not ideal locations for communication; using wired communication increases the risk of entanglement, while obstructions, use of radio channels by emergency services, and background noise can make wireless communication unreliable [10, 41]. In cases of terrorism, radio channels may be suspended to avoid triggering secondary bombs [57]. UARs could be used as mobile communication relay stations, autonomously altering their position to ensure maximum connectivity between areas in the USAR site and the Command Station [32, 57]. Sometimes there will be a need for a single point to be continuously linked to the Command Station such as when delivering a real-time video link to a trapped victim, while sometimes only sporadic updates are required such as a UAR's current position, fuel levels and current tasks; hence each mission task has its own communication requirements.

2.2.5 Environmental Monitoring

Once hazards are identified, some will require monitoring to reduce the risk of rescue workers being exposed to these hazards [14, 57]. Conditions that need to be monitored include harmful or flammable gases, oxygen richness, CO_2 emissions (which can also be used to detect victims), structural instability, and fires. Note that currently, USAR rescue workers will not enter a building that is on fire [27]; UARs could be deployed, however. When the local conditions are deemed safe, the UAR can alert rescuers that it is safe to continue [14]. Hazard data needs to be delivered to the Command Station in real-time where possible. Automating this task would free up certified rescue workers to take part in the actual rescue. Tasking additional workers to the rescue will increase the victim survival rate while UARs monitor conditions using a wide range of sensors, providing continuous, accurate environmental monitoring round the clock without fear of fatigue [14, 41, 57].

2.3 Sharing Data

USAR operations often take place in noisy, cluttered environments where saturation of radio links mean that communication links are only available over short range. Even then, channels are to be considered unreliable at best, with the possibility of a total communications breakdown which must be understood and dealt with [69]. As such, UARs deployed in USAR must be capable of acting autonomously as much as possible, since teleoperated systems rely on constant communication via reliable channels. Autonomous systems that operate without any communications link are an alternative when teleoperation is not available; reduced task effectiveness as a result is acceptable [79]. To reduce the need for dedicated operators and communication links, interim layers of autonomy can be used, including fully autonomous, shared control and teleoperated [12].

2.4 Communication Requirements for USAR

There are two different communication requirements that need to be fulfilled by USAR UARs:

- Reachback: a method of streaming data from the 'remote' team (i.e. the Command Station) to a 'local' team (i.e. rescue experts who are geographically distant). The local team performs data analysis for the remote team, freeing the remote team's resources and manpower to concentrate on their main task, such as search or reconnaissance.

- Inter-node relay: UARs sent into a USAR environment can utilise node mobility and short range communication to achieve the same effect as a larger, more well-connected network. For example, a team of UARs can form a communication chain to provide a real-time video link between the Command Station and a trapped victim. Alternatively, they can cover a larger area in the form of a delay tolerant network.

2.4.1 Reachback

UAR teams operating in USAR generally have little or no infrastructure, and therefore are well suited to local ad hoc networks [9]. However, there is a frequent need for 'reachback' to a distant command post, data processing facility or to some experts (the local team) who can make decisions based on the incoming data regardless of the UAR's location [32]. This is shown diagrammatically in Figure 2.1. For a mission to be responsive in real-time requires nearly constant

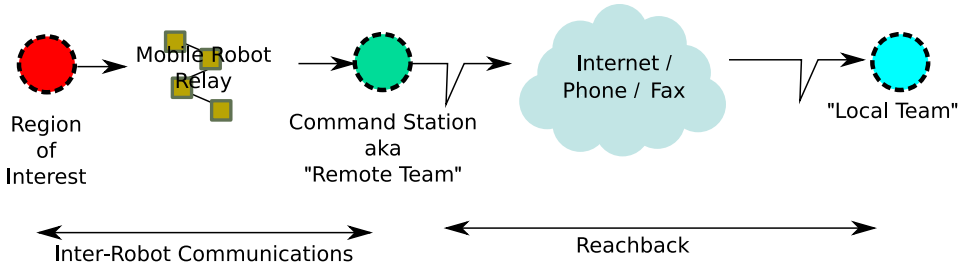


Figure 2.1: Reachback

communication between the local team and remote team. The current practice in search and rescue is to establish reachback via internet, phone or fax; this requires validation by the experts, which disrupts them from their main task [32]. While an ideal scenario is that the remote team will have a high bandwidth, secure, dedicated connection to the local team, in reality the remote team often have limited communications, slow data connections and could even be without power. In addition, many UARs might lose connectivity for short periods of time due to outages [32]. Reliance on bandwidth may be misplaced, therefore setup should maintain a strategy to utilise it effectively [32]. External networks such as the Internet should not be relied upon for any mission critical services [3].

2.4.2 Inter-Node Relay

The US Department of Homeland Security Science and Technology Directorate issued a 'Statement of Requirements for Urban Search and Rescue Robot Performance Standards', which included several communication requirements [53]. One was the desire to extend communication beyond line-of-sight, which could be provided via multi-hop relays, in order to project awareness into, or out of, the collapse. There will be certain tasks that require a quality of service guarantee for some duration of time, such as when providing a real-time video link. This will require a dedicated team of UARs to provide a communication relay between two points. If any of the identified hazards require continuous monitoring then this can be dealt with once the reconnaissance is complete by flagging them as a region of interest [4]. Once a victim is discovered they can also be flagged as a region of interest, and the UAR should alert other UARs or human rescuers so that the victim can be recovered. Controlling a UAR remotely is desired by operators, but teleoperation has several disadvantages including the need for one controller per UAR and a reliable and continuous remote communication link [79]. These teleoperated UARs can therefore be treated as a mobile 'region of interest'. In order to maintain communication, each region of interest must have connectivity to the Command Station. This is a well-researched topic: a mobile sensor network

was developed to explore and find target objects using a controlled deployment of static nodes to provide coverage as the UAR explored the environment [38]. A similar system aimed to maximise area coverage while occasionally moving to a target; complete coverage is not possible without moving around all nodes [6]. An exploration system by Rooker and Birk used multiple UARs to weigh up the benefits of exploring new space against the deficits of communications loss. In this scenario, any two UARs could communicate via any number of intermediate UARs; there must always be a path to the base station, and UARs aim to find the shortest path between any two points to route messages. The maximum range that can be explored is the number of UARs multiplied by their communications range; however this is quickly decreased by obstacles [74]. The position of a single UAR may be constrained by the actions of others (i.e. if that UAR moves then another will become disconnected) [67]. If link quality or available bandwidth drops below the level required for reliable end-to-end communication then UARs automatically reposition; this may result in the system being unable to provide adequate service [39].

Nguyen et al [61, 64, 62, 63] investigated the problem of using a group of co-operating mobile robots to explore an unknown environment under limited-range communications. They show that radio-frequency (RF) communications are most desirable, the same frequencies that are most likely to be unavailable, as seen in Section 2.5.1. Their model has a single exploratory robot investigating complicated interior environments such as buildings, bunkers or caves. The lead robot has a group of four relay nodes that act as slaves; their role is to ensure that there remains a continuous high-bandwidth digital RF communications link from an operator (typically outside the structure) to the lead robot. Nodes are deployed - then later reclaimed and redeployed - entirely autonomously as and when they are needed. The authors work under the assumption that digital radio typically operates on line-of-sight links. The authors implement and test various deployment strategies in a Player/Stage [33] simulation before choosing a convoying method as the fastest, despite the need for four dedicated slave nodes. Nodes are deployed when signal strength drops below a preset threshold; each node has complete link information to make this judgment autonomously, and is able to determine paths to the operator where possible as routing tables are updated whenever links change. Nodes that fall out of the quickest route from exploratory robot to operator can be reclaimed, navigating back to the operator for future redeployment. In experiments on real robots, nodes would autonomously stop just beyond line-of-sight; using this method, the authors were able to navigate the exploratory robot via real-time teleoperation through an entire underground bunker, well beyond

the range that a single link could span. Nguyen et al demonstrate a solution to the problem of maintaining a guaranteed high-bandwidth RF link; this has various uses in USAR, such as allowing rescue workers to communicate via video and microphone with a trapped victim. However, as with other systems, their method does require dedicated slave nodes and has a maximum range; there is also no discussion about the time that the system takes to deploy, nor what speeds the robots can achieve whilst exploring.

Atay and Bayazit [4] created a method for distributed computation of the placement of mobile robots in order to cover specific regions of interest (ROI). Each ROI needs to be within the sensor range of one or more robots, and the algorithm attempts to maximise the number of ROIs covered, the number of robots that are within communications range of each other, and the total area explored. Two robots are capable of communicating if they are in an unbroken chain of robots that are all within communications range of one-another. Using input data such as the speed and location of each robot, location of ROIs and obstacles, each robot calculates an optimal placement for itself and all neighbours; these directives are shared and the option with the highest local utility is chosen. The result is similar to a centrally optimised approach, but 400 times faster to compute, making the system scalable. However there is no discussion about the priority of targets, nor what happens when there are insufficient robots to cover all targets. There is no mention of needing to send data from the targets anywhere for processing, so it is not clear how connectivity is maintained nor what latency is typical. The work of Atay and Bayazit could be implemented in a USAR operation where a list of hazards are treated as ROIs, and mobile robots could cover them so long as they are deployed in sufficient numbers, but there are no guarantees of any end-to-end links to cover all ROIs. Therefore some kind of mobile message passing system is necessary.

2.5 Opportunistic Communication

Communication relay has shown to be effective when there are sufficient relay nodes available. Without sufficient nodes it is simply not possible to guarantee a link beyond a certain range. Most of the time that UARs are in the USAR 'hot zone' they will be pursuing their own goals such as reconnaissance or search; in these circumstances, communication is a secondary objective for the UAR. A method of communication is required which will allow data to be shared between UARs, and therefore to filter back to the Command Station without having a

significant impact on the ability of a particular UAR to pursue its primary task. For example, three UARs could be tasked to perform a perimeter check to ensure that no-one enters the hot zone, to perform an area search to look for hazards or victims, and to provide a medical check of a casualty. Each of these UARs will, during the course of their task, generate new data items that will be of use to the Command Station, but which might not be urgent enough to warrant the UAR moving away from its current task (i.e. mapping data). Whenever two UARs come within communications range of one another, they can exchange data. This will allow UARs to make use of each others movements in order that data can be spread and filter back to the Command Station without having an impact on the primary goal of any UAR. This is a classic application of a delay tolerant network, yet because in this scenario the contacts between nodes are unknown, this is termed an *opportunistic network*.

2.5.1 Communication Methodology

It is difficult to predict the effects of unknown environments on communication measurements, and there are no guarantees that measurements in testing will reflect actual mission performance [39]. Most problems occur due to noisy channels and bandwidth constraints [81]. Limitations in the communications channel can reduce overall performance via time delays and data loss [76], which can lead to a decrease in successful mission completion and verification [55]. Radio-frequency links are desirable but suffer from interference and attenuation problems, and are actively discouraged in RoboCup Rescue because their frequencies are already strained and unreliable from within buildings [41]; spectrum digital systems overcome these problems but operate at much smaller ranges [61]. The best frequencies for penetration of structures are those most likely to be saturated by emergency services or media during an USAR operation, limiting the range that a single link can span [14]. A move away from radio signals towards protocols such as 802.11 and 802.16 has been recommended [72]. In systems where communication links are vital, UARs must monitor the quality of links to their neighbours and move closer if the link drops below an acceptable threshold [39]. Because transmission ranges of a UAR are limited [9], any group of UARs has a maximum range which is dictated by the number of UARs and their individual communications range [89]. It should be an assumption of all UAR teams that UARs might not be able to communicate directly, and that data may need to be relayed via a third-party [70]; all UARs should be able to communicate with each other if there is some path between connected UARs that links the two [4].

2.5.2 Architecture

In designing an architecture it is important to note that we are interested in communication between mobile UARs when they are outside the communication range of a Command Station. This implies a decentralised, delay tolerant approach. Network components (i.e. UARs) will have no knowledge of global topology; they will only know about connections in their own neighbourhood [49], and will exchange data on a peer-to-peer level. Each UAR in a peer-to-peer system is treated as an equal and can share information and resources freely [91]. Vehicle movement and changes to environmental conditions leads to rapid fluctuations in network topology, creating a need for self-configuring, self-sustaining networks of UARs with location-independent architectures [9]. UARs will come in and out of communications range causing 'network merges' and 'network breaks' [21]. Each connection between UARs should be viewed as a short-lived network; each time a network forms, routes may have changed as UARs change their relative positions [35]. A minimum hop path approach is not sufficient because the computation of the minimum-hop path cannot be completed as exact routes may not be known [9]. Each node has limited range and can communicate directly with other nodes in range or indirectly with other nodes via multi-hop relay [48, 70]; throughput falls dramatically with each hop, while delays increase [9]. Routing messages between UARs is not dissimilar to routing via static routers. Traditional routing strategies cannot be used because the network topology is too flexible [81, 89].

The entire system should ideally appear to be a centralised and highly-available data source despite breakages in the network, which may be deliberate (such as UARs moving out of range of each other) or accidental (such as a communication failure) [26]. As the team size increases, bandwidth issues become more important; as more UARs are added, there is a drop both in bandwidth available to each UAR and also in overall network throughput as over-saturation leads to low-level packet collisions. This might be due to a UAR's sensing bandwidth exceeding the network bandwidth [39]; hence bandwidth should be conserved where possible.

2.5.3 Choosing which data to send

All data sent from the remote team to the local team need to be set in context where possible otherwise it can be confusing for the local team to understand. This includes having all data in a standard, non-proprietary format with annotation, timestamps and location. All data should be encrypted as victim data can be sensitive, and should be designed to allow for automatic parsing [32]. Networks should only exchange raw data, with any human-computer interfaces carried out in

secure environments [2]. Individual nodes may lack the storage capacity to hold all of the data that they can sense, so they must be able to offload data to specialised storage facilities; one such example mimicks a traditional client/server architecture [26]. In small groups, global broadcasting is sufficient and has the added advantage of making all information available to the entire group [76]; flooding the network with data in this way is effective for decentralised strategies, but not centralised ones [21]. Broadcasting sensory data to as many UARs as possible minimises the propagation time for that data to reach the intended recipient, while also increasing data redundancy [89]. Sending generic data types may be the best approach for reusable systems [55]. The use of RDF triples to classify data in conjunction with inference rules used in XML can provide context [77], but should not be relied upon to convey meaning beyond those definitions given by operators [17, 60].

2.6 Data Propagation

In ad hoc or delay tolerant networks, there are many methods used to ensure that data can pass through the network quickly to its destination. The method chosen depends heavily on the nature of the particular network; for opportunistic networks, with their low likelihood of maintaining links with other nodes, it is vital to take advantage of each and every opportunity to exchange data. Flooding mechanisms have been deployed successfully in these scenarios [28, 44, 83]. Networks of UARs can be treated as Active Sensor Networks (ASN) with mobility [46]. The ASN approach is to store all data at each storage node in order to give redundancy and therefore robustness against failures, but this creates high storage demands in large environments [49]. All incoming data is assimilated and processed before being communicated on, so only a single message is sent out to each UAR within range regardless of the number of incoming messages. This ensures that the system can scale indefinitely [49]. If the channel between two UARs is suspended (they are out of range, for example) then data simply accumulates until the channel is reopened and the new data is sent in a single message [49]. Replication of data is absolutely necessary to ensure access to common data for users who may become disconnected; however, replicated data can quickly become inconsistent [26]. Using a data propagation scheme ensures that eventually all copies of data will converge to the same state where there are no updates, but at any given time it is highly likely that inconsistent results could emerge. One method involves session guarantees to alleviate this problem, forcing data shares to be collected within a given time frame in order to be useful; another involves classifying data as committed or tentative and allowing other UARs the choice of

whether to deal with tentative data or not [26].

2.6.1 Replication-Based vs Coding-Based Mechanisms

Yu et al [90] describe the nature of opportunistic networks, characterised as any network where contacts are intermittent, where disconnection and reconnections are common, where link performance is highly variable, and where an end-to-end path between message source and destination may never exist. This means that traditional MANET routing cannot be applied. The authors describe how routing schemes in opportunistic networks are based on either message replication or message coding, and that replication schemes tend to perform better when networks are well connected, but are less robust when connectivity is poor. Coding-based schemes reduce the number of transmissions by splitting data up and replicating it, so that only a given percentage of data are required in order to reconstruct the original message. The authors then describe their algorithms for better utilisation of each opportunity to send data. Decentralised Data Fusion (DDF) [30] is a method of combining data in a decentralised system where each node has no information about the global topology, communicating instead on a strictly peer-to-peer basis. Scalability is ensured by performing operations on data locally and then sending a single outgoing message containing all relative updates; thus the amount of information in each message remains constant. When a communication channel is suspended for a period of time, data are accumulated locally and then sent as a single message, thus proving robust to communication failures or changes. Experimental results show that DDF results in the same state estimates as when information is processed centrally, but is faster and infinitely scalable. Coding mechanisms rely on some degree of replication, and can be built on top of a replication scheme. As such, they are beyond the scope of our work, which will focus on creating an efficient replication-based data exchange mechanism.

2.6.2 Replication-Based Schemes

Flooding mechanisms are mathematically proven to be the fastest possible method of spreading data [5], but use a lot of resources. Therefore most replication-based schemes attempt to reduce the number of duplicate packets sent. Roumeliotis and Bekey [75] performed research into mobile robot localisation, using observation and odometry for high fidelity results; all data are exchanged but incoming data are filtered so that only required data are processed. This technique still uses a high amount of bandwidth, but if the idea of using filters were applied by the sender instead of the recipient, bandwidth could be conserved. Rehmani et al [71] note that flooding should be avoided when there is only a single channel per

device, flooding should be avoided to limit overhead; otherwise, new forwarding mechanisms are required.

Vahdat and Becker [86] propose the 'epidemic routing' message passing system for ad hoc networks, where a number of robots move randomly through a bounded environment. Whenever two robots, A and B, pass within communications range of each other, they exchange data buffers in the following way: A sends a summary vector (SvA) of all onboard messages to B. Once received, B calculates the difference between its own summary vector SvB and SvA, then generates requests for missing data. Comparing two - potentially very large - data sets can be a time-consuming process. An optional acknowledgement message can be introduced which will filter back through the system to notify the originator that the message has been received, but the authors do not mention where in their protocol this occurs. Experiments show the validity of the concept, along with delivery rates and latency for 50 nodes travelling in a 1500m by 300m environment for a variety of communication ranges. There is no image of the environment that they tested their algorithm in, but it would seem a good starting point for a data propagation scheme for mobile UARs in USAR, especially since the epidemic scheme imposes no movement restrictions.

Work has been done by Harras et al [34] to improve the efficiency of epidemic routing via controlled flooding. They introduce a time-to-live (TTL) for each data item, limited the number of times that a particular data item would be forwarded (times to send - TTS), and introduced a retransmission wait time (RWT) inbetween transmissions. By treating the original datum as a 'virus', they also propose a 'cure' message that can be introduced once the destination node has received the intended message, which will force all nodes carrying that message to prevent future transmissions. The 'cure' message is only propagated when a UAR carrying it detects the transmission of the corresponding message. Results in simulation when sending a single message from a single starting point to a single destination show that the complexity of a TTL results in added delay, TTS proved to have no impact for sparsely connected networks as the limits were not reached, and RWT attracted significant delays. The 'cure' method was seen to reduce traffic by up to 60% while having no impact upon delay times. The authors also proposed a method to reduce flooding whereby individual nodes would only propagate data probabilistically; i.e. whether or not data was shared had a 50% probability.

Tower and Little [85] use a method called 'Active Curing' by Ho et al from an unpublished paper; ACK buffers (i.e. any data that have been received and

acknowledged by the destination node) are shared and updated first. Secondly, a summary of all unacknowledged data onboard is sent. Nodes are then able to request any missing data, which are then sent. Zhao et al created the Message Ferry [92], where special nodes act as 'ferries' that have a predictable route through the environment; other nodes can then schedule a rendezvous with the Message Ferry in order to ensure that data can be offloaded. This ensures a regular mechanism for propagation from the data collection area back to the destination; one of the potential problems that would restrict the effectiveness of other systems is where there is a 'mobility gap' that prevents nodes from exchanging data, such as where there are two small groups of nodes that do not interact. The disadvantage of Message Ferries is that it requires dedicated nodes which must know the environment they are travelling through and whose movement patterns are dictated by the need to provide connectivity. Similar systems have been created by Chatzigiannakis et al [16], using semi-dedicated nodes called 'runners' which follow a random walk to provide connectivity where none might otherwise exist in order to reduce message delays. While the runners do not have a dictated movement pattern, this strategy still calls for dedicated nodes. Neither technique specifies exactly how message exchange works other than attempting to increase the number of node interactions. This concept of a mobile message carrier has been extended to help rural communities in Finland to maintain a large scale delay tolerant network where mobile relays in the form of motor vehicles provide intermittent connectivity to other communities and to services provided by the Internet [28]. Another system uses the motion of vehicles on roads to overcome a lack of connectivity and provide a delay tolerant service [19]. There are systems implemented which use world models [7], past meetings [44] or other methods for determining which movement direction has the highest probability of success [19, 58]. Probabilistic forwarding methods such as these and [34] restrict the number of messages in the system by reducing the number of nodes willing to be part of the system. Restricting where and when data are exchanged could result in larger delays, and assumes implicitly that previous meetings are a good indicator of future ones.

D Nain et al [59] developed Mobile Relay Protocol (MRP) in order to take advantage of node mobility to disseminate messages to mobile nodes when global communication is not possible. MRP integrates routing and storage; if traditional routing attempts fail, then the message is broadcast to all immediately connected neighbours, who in turn will attempt to pass the message on when they make contact with new nodes. MRP makes no assumptions about node positions or movement, but does utilise a hop limit which could prevent data reaching its destination. MRP is ideal for applications where reliability of delivery is more

important than latency, such as email; the authors envision its use in scenarios where there is no knowledge of the nodes' positions or trajectories, and where node movement cannot be easily predicted or controlled. In simulations, MRP was compared to traditional routing in ad hoc networks, and delivered a higher percentage of messages at the cost of increased latency. The authors acknowledge in their conclusions that their algorithm can be improved by preventing duplicate packets from being sent.

2.7 Trade-off between search and communication

Rescue workers who are outside of the hot zone, in the Command Station performing data analysis or teleoperation, and coordinating the rescue efforts, must try to get information about what is happening inside the hot zone. From their point of view, they must wait until a rescuer comes out of the search space, or comes within communications range, in order to provide a situational update. A single robot links directly to the Command Station (CS) and is tasked to remain static next to the exit of the structure; the other robots need to transfer data to this robot, which will then relay the data back to the rescue team in the CS. The time that it takes to get information about victims to the rescue team can be split logically into two constituent parts: the time taken to locate a victim (Search Time - ST), and the time taken to relay that data back outside the structure (Delay Time - DT), which together are termed the Total Search Time (TST). With a limited number of robots able to perform a search, and a limited ability to communicate, whenever a victim is located the robots need to make a decision about whether the robot should continue searching, or halt the search to report the victim's whereabouts to the CS. Continuing the search will decrease ST for remaining victims but increases DT for the found victims; yet if the robot halts the search in order to minimise DT, then ST for all remaining victims will increase as a result.

2.7.1 Search Methodology

There is a surprising lack of information about how buildings are searched in real disasters; discussions with Technical Rescue officers working with the Leicestershire Fire Service [27] indicated that this is because of the variety of search spaces. It is left to the senior officers to make decisions based on experience. Typically, a building might be divided into floors, with rescue workers splitting up to search left and right down corridors, and having a communication channel running up a

stairway. As already noted in Section 2.1.6, once within a room, rescue workers follow the left wall in pairs. It is with these loose conditions in mind that our experiments in Section 6 has been designed. One field that has looked at the search method in depth is that of animal foraging in biology, resulting in some mathematical models that can produce near-optimal behaviours for searching for food items. By applying one of these theories called the Marginal Value Theorem (MVT) [8] to the USAR task, it may be possible to produce a more effective search. The idea behind optimal foraging theory is this: upon entering a discrete area where food items can be found - called a 'patch' - an animal will gather food at a rate defined by the time taken to find food in that patch and the handling time for each food item. As an animal moves through the patch and consumes food, the amount of food will decrease, and thus over time the rate at which food is found will decrease. In USAR, a patch corresponds to a room or void, the animal to a rescuer or robot, and the food items to victims or hazards that need to be located. The rate-of-gain will then show victims found over time. By performing many searches of each environment, and collating data about where victims are found and when, it has been possible to generate a curve showing the cumulative gain over time for any given patch (see Figure 6.7). By collating all patches and normalising the gain over the number of experiments, the optimal amount of time that a searching agent should spend in any given patch ('patch residence time', PRT) can be calculated by taking a tangent to this gain curve. The curve is offset along the x axis by the mean time spent travelling between patches; the longer it takes to travel between patches, the longer the agent should search each patch. The intercept between the tangent and the gain curve gives us the optimal PRT. Dechaume-Moncharmont et al [25] looked at the trade-off between search and communication by analysing whether or not it is worth finding help to deal with extremely profitable patches. Their study shows that waiting for assistance and attempts to share information can be counter-productive, and that it is often more beneficial to search alone.

2.8 Related Work

2.8.1 Cooperative Search

The following are examples of research where cooperating groups of robots with limited communications range are attempting to search an unknown space, and where the authors are examining the effects of limiting communications on the ability of the team to perform their task.

Ray [69] developed a system using the Player/Stage simulation environment [33] where teams of cooperating robots would explore an indoor environment in order to search for a single target object, mapping as they went. Maps are exchanged in order to speed up the search process and to prevent the same area being searched multiple times. When the target object is found, its location is sent to all other robots. Receipt of the target's location switches each robot from 'search' mode into 'rescue' mode, and they all then plot a course to the target. The simulation ends when all robots reach the target. The experiment was run under three conditions: one with continuous communication, one with occasional communication, and one with no communication. Continuous communication proved to be a drawback when more than four robots were tasked together, as they would spend too much time performing communications, and also because of inter-robot interference (such as needing to avoid other robots). This resulted in an increase in time to completion. Occasional communication resulted in a higher number of successful runs, with slightly faster search times than continuous communication, but with higher inter-robot interference and more wasted time as robots covered the same area multiple times. No communications proved the slowest and also had maximum interference. The main drawback in this work is that the model assumes global communication, and there is only a single target so once it is found the experiment is over. With limited communications range a robot may need to leave the target in order to pass its coordinates on, and with more than one target the robots will need to continue searching until they are sure that all targets have been located. These two conditions create a variety of interesting problems.

Burgard et al [13] used a team of mobile robots to explore and map an unknown, obstacle-filled environment. The environment is divided into occupancy grids, each cell containing a probability whether it contains an obstacle or not; all cells start as "unknown", then as they come within sensing range they are assigned a probability related to the result of a scan. An algorithm was developed to force robots to explore the environment, pushing them towards frontier cells by calculating the expected utility of targeted cells in comparison to the cost of travel, and encourages robots to spread out by reducing the utility of cells near an already-assigned target. Penalties for collisions encourage robots to stay in the open. Using the current location of each robot, the algorithm attempts to maximise the expected utility while minimising the cost. This is a centralised approach assuming lossless global communication; when the authors restricted the range of communication then the algorithm was applied on each sub-group. This resulted in a worst case scenario where each robot searched the environment individually, so robots were allowed to exchange a list of previously explored targets to prevent

repeated searches. Experiments showed that a communications range of 30% of the environment size was enough to yield the same results as unlimited communication, but with vastly reduced costs. The robots do not actually search for a specific object, nor do they need to pass any data to a collecting point, robots simply aim to explore new areas. As the communications range was reduced, the results became similar to an uncoordinated strategy; there was no discussion about how many robots and how much interaction are required to achieve an effective search.

Winfield [89] used a group of mobile robots to explore a bounded region, using the mobility of the robots and their ability to form ad hoc networks to pass data around, describing the overall network as a single, 'broken' ad hoc network. In this manner, data could be passed back to a collection point without requiring complete environmental coverage. Without command or control structures and having individual robots move on a random heading, Winfield was able to exploit the mobility of the robots in conjunction with a highly dynamic topology in order that data propagated through the system "in virus fashion". If data are not required in real-time (i.e. the system is delay tolerant) then this method proved to be capable of performing the function of data relay without any maximum distance. Delay times increased as the number of robots decreased. A number of reasonable assumptions were made, including that each robot had ample local storage and that all data transfers could be completed in the available transfer windows. Data sensing and communications are only performed in a specific time frame; this could lead to onboard latency and unnecessary delay. Winfield introduces a novel concept that could be utilised in order to give mobile robots the ability to pass data without affecting their movement patterns and therefore without significantly impacting their primary mission. However, these experiments were performed with large numbers of robots in a relatively small area, without obstacles.

Scheidt et al [78] used stigmergic communications - i.e. indirect communication by altering and observing the local environment and other robots - in addition to explicit communication in order to control a group of mobile robots. By observing other vehicles, obstacles and targets, each robot was able to calculate its own heading based on a belief system; if a target was believed to exist then it would act as an attractive force, while obstacles act as repulsive forces. Explicit communication was performed by broadcasting data periodically without knowledge of recipients; received data was used to update the robot's world model. This allows knowledge to propagate without the need for a continuous network. In experiments, the au-

thors broadcasted a belief into the network, and the robots responded by adopting an appropriate behaviour; this shows that, even without a fully connected network or routing, a team of cooperative robots were able to spread data and react to it. The authors argue that optimal solutions are not only unnecessary in order to achieve an effective solution, but can be counter-productive as individual elements fail.

Speranzon and Johansson [82] developed a cooperative pursuit-evasion game in order to compare the effectiveness of time-triggered and event-triggered communication schemes. Each robot built a map based on its own sensor information, and all robots exchanged their maps in order to build maps with higher confidence; maps were overlaid and obstacles plotted probabilistically. The authors were able to get similar results when using the event-triggered scheme as with the time-triggered scheme, but with far less communication. They conclude that the use of a communication link can be improved significantly by restricting communication to certain events while maintaining the same game results.

Rooker and Birk [74] developed 'communicative exploration', a scheme that deals with the trade-off between the benefits of exploration and the deficits of losing a communication link with a base station. Groups of mobile nodes aim to move towards the 'frontier' (i.e. area of explored space adjacent to unexplored space) by maximising a local utility where rewards are given for the exploration of new spaces, and punishments for hitting obstacles or leaving communications range. Only a subset of all possible moves are calculated due to the intractibility of calculating a large number of nodes. The result is a highly efficient algorithm with close to optimal results. This system does have a maximum range that can be explored while maintaining constant communications with a base station, and there is no discussion about how exploration could occur beyond this range. Also, the local utility is a step-based function; effectively each move is made like the pieces on a chess board. The current velocity of each robot is not taken into account; therefore this process is likely to be extremely slow compared to other exploration schemes.

2.8.2 Opportunistic Networks

Juang et al [44] created a wireless peer-to-peer network called ZebraNet for the purpose of tracking wild zebra in the field. Collars - acting as data collection and communication nodes - were attached to zebra. Since there was no coverage of the vast area in which the zebra live, and because node movement was - from the

point of view of the researchers - random, a system was required to collect data from the nodes. Some fixed base stations were used to collect data; in addition, the researchers flew a plane over the herd to collect data from the collars and base stations directly. In order to do this effectively, data are shared between nodes whenever zebra come within communications range; this data is exchanged using a flooding mechanism. Communications range between collars was kept deliberately low to conserve battery life. The authors go on to describe their experiments in simulation and in the field, and how they attempt to use historical data on zebra movement patterns as an indicator of future movement in order to selectively forward data, thus reducing the number of data replicated. This means that a single sweep by a research plane should be able to collect data from a large group of zebra. The collection period of ZebraNet is more than a month, and individual nodes are extremely static compared to other systems; it would take an individual zebra over 35 hours to cross the width of the data collection range moving at full speed. Adopting an opportunistic method of data propagation means that some of the USAR UARs will not need to move back within communication range of the Command Station in order to pass data on; it will travel indirectly via other UARs. The main difference between our work and ZebraNet is that ZebraNet has extremely long duration and can tolerate very high latency; data are collected over a month, far beyond the length of time that USAR operations will extend. Node movement in ZebraNet is extremely slow compared to UARs in an USAR environment, with latency of up to a month; in our experiments, latency should be in terms of minutes. In our experiments, the Command Station is also static (compared to the research plane in ZebraNet), mainly to remove another variable from our results, but in a real USAR system the role of Command Station could easily be filled by another mobile UAR which would act as a relay to the real Command Station. Su et al [83] developed a system that utilised human movements around a college campus to form a network; user movement proved to be far from random. Data collection required students to bring the devices back to the researchers; this method is currently the norm with UARs (i.e. return to base and offload data) but opportunistic methods of data exchange might be able to pass data back to the Command Station via other UARs, which could allow UARs to stay in the field longer.

2.9 Summary

Urban Search and Rescue operations are launched in a wide variety of circumstances and environments, many of which are hazardous to human rescue workers who are easily fatigued and generally overwhelmed by the sheer scale of the task at hand, and who can easily become victims themselves. Certified rescue workers are in short supply, and organising and executing a rescue operation in a chaotic environment is an extremely complex and difficult task, both mentally and physically. To aid this, a standardised response mechanism is used at all USAR incidents. A local command centre is set up where the USAR operation is coordinated, then data is sent for analysis at a remote centre which allows certified rescue workers from all over the world to assist.

Decisions to commit rescuers to a rescue site are left to experienced personnel; there are no standardised decision-making tools. Rescuers gather as much information as possible from eye witnesses or other victims prior to entry in order to create a good picture of where casualties might be. If rescue workers are sent in then the first stage is assessing the area for hazards, creating maps of the environment and the locations of any hazards and victims. Some of these hazards will require continuous monitoring in case a change in situation leads to a danger to rescue workers. Once the area has been assessed, the search for victims can begin. It is vital that survivors are found quickly, since survival rates drop rapidly the longer people are trapped. Once inside a structure, rescuers follow walls, using touch and short range sensors to find casualties. Victims can be suffering from a wide range of problems and so medical assessment is urgently required; unfortunately, victims are often beyond reach, which requires the penetration of rubble piles. Rescuers use a wide variety of technologies to assist in their search for survivors, and then relay data back to the local command centre. This must be done securely due to the sensitive nature of data about victims. The environment can mean that communication is made very difficult or even impossible at times.

UARs can be used to assist in most of these tasks; they do not require training and can be exposed to dangers that humans cannot. Since the number of rescue workers allowed into the USAR site is usually a bottleneck, being able to deploy UARs will greatly ease the burden on the rescuers. Groups of UARs should be able to cooperate their efforts to achieve more. There are a whole host of technological problems that must be overcome before UARs can be deployed in USAR environments.

The communications constraints of USAR make for an interesting and challenging research area, since many of the other tasks such as search, reconnaissance or environmental monitoring, all have their own communication requirements. It is with these issues in mind that we look at developing a method for inter-UAR communication for use in USAR. We assume that there is no environment-wide communication network in operation, which is often impossible. USAR sites are chaotic and not ideal for communication; wireless links are often reduced to line-of-sight operation. Dedicated communication relay using mobile nodes, sensor networks or teams of cooperating UARs is a well researched topic but, in a complex USAR environment, can require a large number of UARs to provide a single point of interest with a communications link. Traditional routing mechanisms fail due to the sparsely connected nature of these ad hoc networks; as a result, node mobility and data replication are required for effective message propagation. It is important that these relay mechanisms work without knowledge of node movement; prior to the rescue, it may not be possible to know the environment, number of nodes or where they will be tasked. These kind of sparsely connected ad hoc networks where node movement is either random or unknown are called opportunistic networks. UARs deployed in USAR environments, with their limited communication ranges and individual node mobility, are ideal for the application of opportunistic networks.

We will now look at opportunistic networks in USAR environments by implementing various search strategies and analysing the impact of opportunistic data relay in terms of the effects on the search, the time taken to relay data back to the Command Station, and the costs in terms of data sent. Creation of this mechanism will allow UARs to exchange data opportunistically via inter-node relay without significantly impacting their ability to pursue a primary goal. While the use of deliberate data relay by the likes of Nguyen et al [61, 64, 62, 63] show how node mobility in conjunction with short-range communication can be used to provide a link between two points, Winfield's work [89] shows that random movement of nodes is sufficient to allow data propagation, given enough nodes, interactions, and latency. Whether or not this method will be able to provide an effective solution in a USAR scenario remains to be seen; Winfield used large numbers of nodes in an open space without obstacles. In this respect it is similar to ZebraNET, created by Juang et al [44], an example of a functional opportunistic network in the field, where zebra act as 'nodes' with pseudo-random movement patterns. Data is able to propagate when zebra socialise, despite the vast range of habitat and the low communication range used. The mobility of zebra in their environment is far lower than a UAR in a USAR environment, and the number of nodes are also quite dif-

ferent, so the expected latency is also expected to differ. There is little information about how often nodes in ZebraNET exchange data, but as zebra are naturally a social animal, it may well be more often than UARs which will probably spend most of their search time alone. This will also have an effect on the results in USAR enviroments. Other methods must therefore be used to improve propagation efficiency. Su et al [83] also implemented an opportunistic network, this time using the movement of students on a college campus. The results showed that most students use main thoroughfares and data could be exchanged in meeting places such as canteens or lecture halls; again, this is similar to the herding nature of ZebraNET, but not applicable to USAR. Therefore, experimentation is required to see whether the use of opportunistic communication is viable in USAR operations.

There are a variety of data exchange protocols that are employed by mobile nodes in ad hoc networks, generally based around modifications of a flooding mechanism. It is vital that this mechanism is efficient to avoid wasting energy sending data that is unnecessary since saved battery life results in longer run-time for each UAR. We look at the application of these protocols in a USAR simulation, with a view to creating an efficient protocol. Winfield [89] used a pure epidemic replication method, but Juang et al [44] were able to use historical data about zebra movements to improve the data propagation efficiency; this is possible in social herds that use regular watering holes or feeding grounds, but not applicable to UARs exploring an unknown environment. A range of replication-based propagation schemes were discussed in Section 2.6.2, and these will be analysed and compared in Chapter 5.

During any rescue, the rescue workers outside the structure (in a Command Station) must build up a picture of what is inside. When it is not possible to provide communications to the entire environment, the Command Station must rely on robots moving within communications range to get updates. Use of an opportunistic network of UARs can help to reduce latency, and therefore victims will be known to the Command Station earlier, resulting in higher survival rates. The overall search time of an environment is made up of both the time taken to perform the search - which we refer to as the search time, ST - and the time taken to get the data back to the Command Station - which we refer to as the delay time, DT. There is a tradeoff between reducing each of these constituents, with the aim of reducing the overall total search time, TST. Ray [69] demonstrated cooperative search in simulation, and showed that continuous communication was not necessary to be effective. However, his experiment only searched for a single target. This is a useful starting point, but can be expanded to include multiple

targets, different numbers of nodes, and different search methods and movement patterns. Introducing multiple targets in an environment where there is no global communication method also introduces the issue of a trade-off when a target is found, as to whether the node should continue to search, or communicate the findings back to the Command Station. Burgard et al [13] analysed the trade-off of maintaining communication link or exploration, and showed that global communication is not required to achieve a near-optimal solution, this is the same conclusion as Rooker and Birk [74], studying a similar trade-off. Both sets of experiments tried to maintain communication links at the expense of exploration - the algorithms were weighted to ensure that communication was a priority - and so there is no discussion of the effects of following this strategy on the time it takes to explore the environment in full. Scheidt et al [78] implemented coordinated robot behaviour via the use of broadcast 'beliefs' on a real test-bed of mobile robots. This is useful, as it indicates that the conclusion of taking locally optimal decisions can lead to an effective strategy in a real environment where global communication might be impossible. We will examine this trade-off in more detail in Chapter 6.

Chapter 3

Experimental Environment

3.1 Player/Stage

Gerkey et al [33] created the Player/Stage project, an Open Source robotics tool which aims to simplify development by being language and platform independent. Client libraries are available in C, C++, Tcl, Python, Java and Common LISP. Player is a socket-based robot-device server that provides a logical interface to individual robot sensors via a TCP socket, enabling remote access. Player executes on any machine that has a network connection to the robots and their sensors, and offers connections to these devices; clients connect to Player and communicate with these devices by exchanging messages with Player. Multiple clients can be supported simultaneously. Stage is a multiple robot simulator, offering a 2D simulation environment where devices are accessed through Player as though they were real robots. Stage can simulate hundreds of robots simultaneously, and is highly efficient and configurable at the expense of some accuracy. Simulation via Stage enables faster development and reduced costs to developers when compared to developing directly on robots. Once code has been tested in Stage, it can be directly ported onto real robots if Player is also installed, requiring little or no code modification. However, simulations in Player/Stage cannot currently be executed in faster than real-time, hence collecting results from thousands of experiments takes a large amount of time.

When developing for Stage, an environment needs to be created. This is done in the form of a *cfg* configuration file that is used as a parameter when running Player; this configuration file contains details of all of the devices that can be accessed through Player, such as a server, robots and their sensors (and corresponding port numbers), and the Stage environment *world* file. This world file lists characteristics of the simulation environment such as which map to use and

what size it will be, and also pulls in other configuration files that correspond to the types of devices listed in the cfg file. Why these two have been split is not clear. This means that, in order to create a robot in Stage, there has to be an entry in the world file corresponding to an entry in the cfg file, with the addition of the starting coordinates and robot colour. This design feature is unfortunate because it means that an environment needs to be defined for a specific number of robots; it is not possible to use a generic environment and change the number of robots using code written in the client libraries. In order to alter the number of robots in an experiment, it is necessary first to define a new cfg and world file pair. Additionally, despite having their coordinates specified in the world file, when executing code the coordinate system starts all robots at (0,0); in order to remedy this, it is necessary to manually define the intial position and pose of each robot within the client code, replicating data in the world file. These awkward design issues aside, Player/Stage provides an excellent platform for distributed robotics simulation. Player/Stage has been chosen in preference to other systems because it is open source, freely available, and currently supported. In particular, the RoboCup Rescue Competition [http://www.robocuprescue.org/] uses the USARSim robotics simulator [80] for their rescues, but this is geared to generating extremely realistic building collapses using a physics engine, and for testing either locomotion and localisation within voids, or looking at disaster management strategies. Using Player/Stage removes a level of complexity and a large number of variables from an already complex problem.

3.2 SEIC Laboratory

Much of the initial robot behaviour development at Loughborough University was performed at the System's Engineering Innovation Centre (SEIC) laboratory, where a testbed is used for development and testing of autonomous systems. The laboratory has six Koala [40] robots, which are highly mobile and easily programmed for rapid prototyping. Koalas are 30cm x 30cm, with six wheels as seen in Figure 3.2 and are fitted with a Hokuyo URG-04LX laser rangefinder [36]. Koalas communicate wirelessly via 802.11 on a simulated peer-to-peer model; in reality, all communication is performed via a server, but from the point-of-view of each robot they communicate directly. Development in the SEIC laboratory helped to expose some bugs - such as being able to filter noise from the lasers - that was not required in the 'perfect model' provided by Player/Stage, leading to more robust behaviours. Early on in the process of developing this thesis, some development work was done on the Koala robots, and the Player/Stage simulations have been created with the express aim of porting the code directly to the

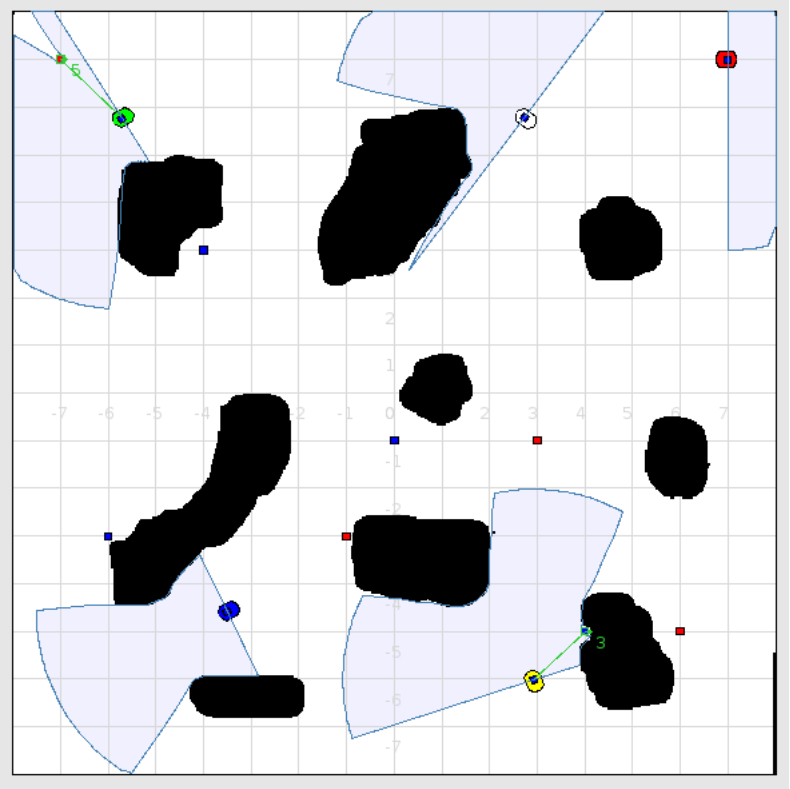


Figure 3.1: Simulation environment, running in Player/Stage

Koalas. As such, the physical characteristics used in our Player/Stage simulations have been chosen to mimic those of the Koalas.

3.3 Robot Localisation

Localisation of mobile robots is an unsolved research issue. It would not have been possible to perform the experiments that will be described in later chapters without having some kind of localisation mechanism, yet to develop robust localisation is a research issue in its own right; even methods such as visual SLAM are insufficient, and novel techniques are required. One such method is the distribution of RFID tags which can be used as reference points for localisation based on line-of-sight [45]. In order to be able to concentrate our research on the search methodologies and communication strategies required for USAR, we have made use of the internal odometry within Player/Stage. Odometers are internal sensors that keep track of which direction robots move and at what speeds, then use this internal count to keep track of the robot's position within the world model. In a simulation, this process is error-free; however, in real robotics, even highly sophisticated robot systems are subject to errors - called *slippage* - as a result of less-than-perfect contact with various floor types such as shiny floors, carpets or

Figure 3.2: Koala robot

gravel, plus the contours of terrain. This means that our simulation relies on a perfect odometer, which will not cause problems within Player/Stage but could prevent our system being directly ported to a robotics rig unless

3.4 Behaviour Development

All code required for the experiments in Sections 4, 5 and 6 has been written in C++ for Player/Stage in a modular fashion; behaviours can then be combined to form more complex ones. Because each experiment has multiple robots, the ability to quickly create behaviours running concurrently on multiple robots from a single script was a key aspect during system design. Behaviours developed by me for these experiments include:

- *Random walk with obstacle avoidance*: each robot is fitted with an onboard Hokuyo URG-04LX laser [36], with a 4m range and 180° field of view; this is to mimic the setup used in our laboratory, with a view to porting our algorithms directly onto real robots. Readings are taken from the laser and fed into an algorithm which calculates angular and linear velocity in order to avoid obstacles; robots will therefore tend to stay in open spaces. If there are no obstacles then the robot will proceed forwards. There are special procedures that need to be followed if the robot becomes trapped, and all velocities are smoothed in order to avoid jerky movement. Because even slight alterations in the laser readings will result in the robot following a different velocity, in an obstacle-rich environment the overall result is a smooth and effective random walk.

- *Navigate to a specific position, regardless of obstacles*: when there are no obstacles, moving to a given coordinate is a straightforward procedure: ro-

tate to face that position, then move forwards until the distance is covered. However, when there are obstacles in place another strategy is required. The method used was to treat the target as an attractive force, and obstacles as repulsive forces. The overall result is to slowly steer to the target regardless of obstacles. When the path ahead is clear and the angle to the target is small, the robot will continue straight, reducing the time taken to reach the target. This technique uses waypoints for navigation; in Player/Stage these are coordinate points using odometry, but could be easily be markers in visual SLAM if ported to a suitably equipped system.

- *Data exchange*: calculates when nodes are in range, which data need exchanging and performs the exchange. There are two methods for data exchange; one is based on distance between nodes, the second is based on line-of-sight, based on literature discussed in Section 2.4.2.

- *Check targets*: check for targets in range; if a new target is found, add a new data item

- *Class creation*: creation of classes that allow multiple instances of robots to be run concurrently using the same behaviours; i.e. identical apart from their ID.

- *Batch Processing*: use of classes allowed the use of batch processes to run experiments automatically thousands of times, with data written directly to file. In this way, experiments could be set up and left to gather data for days or weeks at a time. Experiments in Player/Stage can only run in real-time, and with each run taking between two and eight minutes depending on which experiment is running, performing thousands of experiments meant that data collection took months.

- *Inter-robot communication*: there is little support for communication in Player/Stage; as such, a method for inter-robot communication has been developed. Player/Stage allows the use of a server which acts as a virtual blackboard that all robots can listen to. Only the most recent message is displayed. In order to restrict a robot's ability to listen to all messages, rules were implemented such that each message has a sender and recipient ID, and only if the recipient ID matches the robot's ID does it process the message. Each message also has a message type, which affects the way that message is dealt with at the server side. There are a wide variety of messages: some messages are for the purposes of handshaking, others contain data that is being exchanged or receipt messages from the CS, while others are messages

between robots and the CS confirming that searches have been completed. This solution effectively models a peer-to-peer communication methodology.

The above functionality and behaviours have been put together in order to conduct the following experiments:

- *Search task; Random Walk*: each robot follows an obstacle avoidance strategy that results in a random walk. Stochasticity is present in the form of node movement.

- *Search task; Area Division*: the arena was divided into a given number of sections and the robots followed a fully scripted path by moving to each in a list of specific waypoints, as shown in Figure 3.1, which shows four UARs performing an Area Division search in the 'Search Task' experiment from Section 4.2.3.

- *Search task; Daisy Chain*: one robot followed a scripted search of the entire environment space, while the others calculated the mid-point between their position and that of the next robot in the chain and navigated to this coordinate. The result is a chain of robots, similar to the work of Nguyen (see Section 2.4.2); note that in our experiments, robots are not guaranteed to remain within communications range of one another.

- *Search task; Wall following mode*: when entering a new room, follow the nearest wall edge until exiting the room. This is a technique used by the fire service [22]

All programs have been written as functions to ensure that the code is reusable and easily tested. The overall archiecture has been designed with reusability in mind. All of the different behaviours use the same core functionality; differences in behaviour are due to calling different functions or passing different parameters. Experiments in Chapter 4 use the same communication protocols and three different movement models, implemented via three different functions, each of which calculates the robot's angular and linear velocities. Experiments in Chapter 5 use the same movement parameter, but perform data exchange by calling different functions. Experiments in Chapter 6 use one common behaviour which is passed different parameters according to which method is being implemented. Due to these design features, additional experiments can be run without needing to recompile code, only parameter values need to be adjusted in batch files.

3.5 Assumptions

In our experiments, Player/Stage allows each robot to have an odometer, and for each robot to connect directly to each other's odometers; we have used this to calculate where robots are and used these positions to determine whether robots can communicate. Within Stage, the odometry reads perfectly, and hence there is no slippage. In reality, robots would need to use a beacon to find out whether there are any other robots within range. In Player/Stage, all robots communicate via a central server, so all can communicate at any time unless restrictions are set; in order to create a realistic testbed inter-robot communication has been restricted either using distance (Experiments 4.2 or 5.6), or via line-of-sight (Experiment 6.3).

All experimental results have been averaged over a minimum of 40 runs. Stochasticity is introduced by variations in the response time of the Player/Stage client to various threads, resulting in variability across multiple runs of the same experiment. In experiments where results are plotted with error bars, the error plots plus and minus one standard deviation of the mean.

Physical characteristics:

- Each robot has a Hokuyo laser rangefinder [36] with a 4m range and 180°field of view to its front. In simulation, these lasers do not suffer from noise, interference or failures.
- Movement speed has been restricted to a maximum of 0.5m/s for linear velocity and 180° per second for angular velocity, and a turning circle of zero (i.e. able to rotate in place), in order to match the Koala [40] robots in our laboratory.

3.6 Nomenclature

Throughout the course of this thesis, a number of terms are used. For reference, these have been grouped and explained below:

Command Station (CS) : this is the data gathering point. It can refer to any point that is connected to the Remote Team; in our experiments we treat a single robot as the CS, assuming that it is connected to the Remote Team, i.e. that it is within range of a dedicated communication relay. Therefore we can assume that when data reaches the CS it has effectively been received by the Remote Team and the human operators have access to it.

Local team : rescue workers who are in their own local environment, who can only assist the USAR operation via telephone, fax or internet technologies.

Remote team : the team on the ground, located in the 'cold zone'.

UAR : unmanned autonomous robot, although they could be semi-autonomous or even controlled by humans if within communication range.

Node : either a UAR or a wearable computing device carried by a human rescue worker (or search dog).

Inter-node relay : the process of passing data between nodes involved in the USAR process; these could be robots or human rescue workers carrying mobile data storage devices with wireless connectivity.

Reachback : the process of transmitting data from the 'hot zone' to the remote and then local teams.

Region of Interest (ROI) : any position that requires maintenance of a communication relay; this could be a victim that the Command Station would like to maintain a video or audio link to, or an environmental hazard that requires monitoring. ROIs can be used as input for relay nodes that can then ensure these regions are covered.

Search Time (ST) : the time taken to locate a victim, timed from the point that the robots enter the environment.

Delay Time (DT) : the time taken to get a victim's location back to the CS, timed from the moment that the victim was located.

Total Search Time (TST) : the sum of Search Time and Delay Time, this is the time taken for a victim's location to arrive at the CS, timed from the moment that the robots enter the environment.

Patch Residence Time (PRT) : the time that a robot spends in any one room.

Patch Search Time (PST) : the time taken after entering a room to when a victim was found.

Patch Travel Time (PTT) : the time taken to travel to a given room from the previous room.

Chapter 4

Opportunistic Data Sharing in USAR

4.1 Introduction

In Chapter 2, we saw how mobile nodes could be deployed in USAR environments. Due to a variety of reasons, it is not always possible to maintain communication links through an entire USAR search space. As a result, any deployment should assume that global communications might not be available, and instead should use node mobility in conjunction with short range wireless communication to allow for data propagation. Given the chaotic nature of USAR operations and the short response times required, it is unlikely that the environment in which UARs are deployed would be known a priori, beyond a basic building layout. As such, it might be impossible to predict the movement patterns of UARs once the search operation has begun. These characteristics are ideal for the application of an opportunistic network. We will now investigate the feasibility of using an opportunistic network in a USAR scenario.

4.2 Experiment 1: Search Task

We set up a search task in order to compare various search strategies with and without using opportunistic encounters to relay data, in order to test the effect that data exchange has on the speed of the search task.

4.2.1 Method

Our experiment assumes that the USAR area is too large and the number of UARs too small for constant communication channels to be maintained. Therefore each UAR will sporadically be out of range of others; the idea is that node mobility

can overcome the lack of communication by allowing data to spread throughout the system, as seen in Fig 4.1. From the point of view of the Command Station (CS), all of these UARs will be out of range until any of them comes close enough to exchange data. The aim of our experiment is to see whether, during these periods, UAR interactions will allow data to be forwarded to the CS without relying on each individual UAR to make direct contact with the CS. Whenever any two UARs come within communications range, they exchange messages. Discovery is performed using odometry in lieu of GPS or radio beacons; handshaking messages then establish what data are required by each UAR, and requests are made prior to data transfer. Real USAR scenarios will need a variety of data types; video, images, sound, and more, but in our experiments we only exchange data types such as coordinates. As such, our system has been designed to be generic [55]. If the actual data types are known then the size of each data item can be used in conjunction with the number of sent data from our experiment to give estimates of what the bandwidth and storage requirements might be. We have distinguished between handshaking messages and messages that sent data items, both for this purpose and as additional metrics for comparison of methods. In addition, we do not attempt any routing; all data exchanges are performed by immediately connected neighbours. In this scenario the CS has a fixed location, but could equally be mobile; for instance this scenario could represent a small group of UARs working in one area of the site, with another UAR that periodically moves into communication range of the CS to get and provide updates. Movement speed of each UAR has been set at a maximum of 1.5m/s, reflecting the delicate nature of terrain at a real USAR site; in practice, it is likely that actual speeds will be much slower. Algorithms that run on the UARs must operate in real-time, since off-board processing might be unavailable in the event of a communications breakdown [57]. We have assumed unlimited storage capacity since storage space is generally cheap now. However, if a deletion strategy is needed then data acknowledged by the CS should be deleted first, followed by the oldest data from other UARs. A UAR should avoid deleting its own data where possible, to ensure that there is always one copy in the system for the sake of redundancy [44]. We also impose no movement requirements on the UARs; ideally, opportunistic relay should make use of the current movement of each UAR to ferry messages without having any detrimental effect on that UAR's task other than the cost of sending and receiving messages.

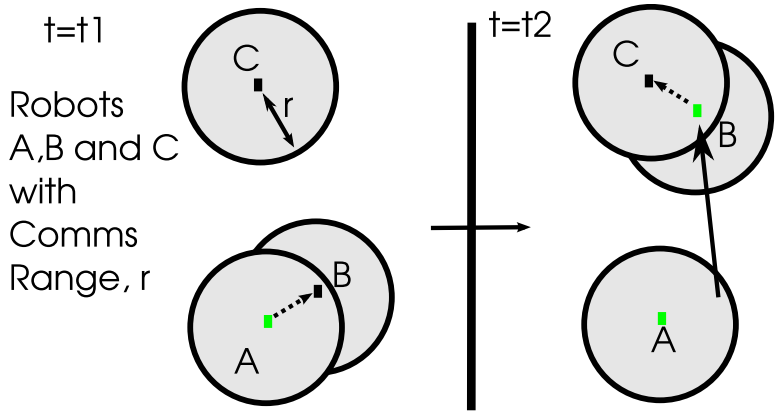


Figure 4.1: Using node mobility to relay messages

4.2.2 Proof of Concept

A simulation using Player/Stage [33] has been developed to see how data filters back from discovery to the Command Station. A single target was placed in one corner of a 16m x 16m grid test arena with obstacles; USAR test arenas are typically smaller, such as 8.5m x 7.3m [47], or 1/8th scale arenas [56] which scale up to approximately 16m x 16m. Four search UARs start in the diagonally opposite corner of the arena to the target, without knowledge of the environment, following a random obstacle avoidance technique based on one developed by Borenstein and Koren [11], and using a 4m laser rangefinder to scan for obstacles. As soon as any UAR comes across the target, its coordinates are recorded. Whenever two UARs come within communications range of each other, any data that they have onboard are shared. An additional UAR, which starts alongside the search UARs, represents the Command Station and remains stationary throughout the experiment. As soon as the coordinates of the target are received at the Command Station, the experiment is complete. The experiment is run with four UARs moving randomly and exchanging data opportunistically, and these results are compared with four UARs who perform a search strategy and return to the Command Station when the target is discovered, but who do not exchange data between each other. Note that we are not measuring the time taken to find the target, we are only interested in the delay involved in getting the target's coordinates back to the Command Station once the victim is discovered; results are averaged over 40 runs; stochasticity is present as a result of timing differences in the interactions between the various threads that connect to the Player/Stage server. In a real USAR operation no UAR's motion will be random, but their movement patterns might be unknown a priori; hence using a random walk to model that uncertainty.

Results (Figure 4.2) indicate that, with a communication range of less than

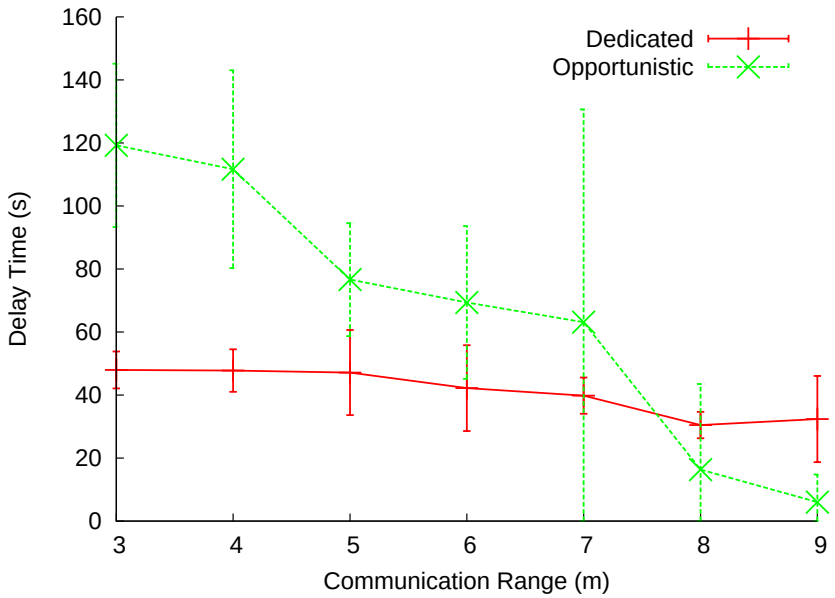


Figure 4.2: Proof of concept results show the delay after finding a target to the target data reaching the Command Station

half the environment size, a dedicated UAR was able to relay data to the Command Station much faster than opportunistic relay, since the network is so sparsely connected. With a communication range of 4m, opportunistic relay with random node movement has lower delay in approximately one-third of experiments than a dedicated and directed node. When using opportunistic communications, latency is highly variable and - due to the nature of random movement - there are no guarantees that data will be delivered. While outperformed by a dedicated relay, the advantage of an opportunistic communication strategy is that no movement restrictions are applied to the UARs, which are therefore free to continue their primary task at all times (in this case, search). The application of an opportunistic communications protocol in USAR could still speed up the delivery of data from within the USAR environment to a fixed Command Station, and potentially remove the need for dedicated communication UARs in all applications where data are not required in real-time.

4.2.3 Search Task

In order to test the feasibility of using only opportunistic communication, a search task is being used as a benchmark. Several search strategies will be used, and both the actual search as well as delay times and communication costs will be compared. Based on the knowledge that structures within voids or collapsed buildings are highly unlikely to have large open areas, a communication range of 4m has been

selected; above this range, no inter-UAR communication is possible; below this range, communication is allowed without failure. Results from the Proof of Concept experiment in Section 4.2 show that opportunistic data exchange is clearly weaker than dedicated relays for a range of 4m, thus ensuring that the results of these experiments cannot be due to having a large communication range. It is assumed that the transfer window is large enough to allow all data to be sent and received without loss [89]. Later we will look at modelling communication patterns that closer mimic real scenarios. Four UARs will be used for the search task, with a fifth acting as a stationary Command Station. Several search strategies were implemented, all using the same method of data propagation.

Area Division The first method uses an area division tactic: the environment is divided into equally sized sections, one for each UAR. All UARs start in one corner of the environment, then travel to their section and cover the ground by snaking backwards and forwards (similar to the 'lawnmower' method used by Andrews [1]), then return back to the starting point to exchange data with the Command Station.

Daisy Chain In the Daisy Chain method (similar to a method by Nguyen [62]), a single lead UAR searches the entire environment alone, with the remaining UARs attempting to form a chain from the search UAR to the Command Station. The result is that sometimes there is a direct connection between the search UAR and the Command Station; when there is not, the chain UARs continually move in and out of communication range with each other and the Command Station in an attempt to deliver data as soon as possible.

Random Walk The final strategy is a random walk, which is the only strategy of the three that does not impose any movement restrictions on the UARs, and we expect that it will not be as fast as the other methods which use a priori knowledge of the environment to ensure that all areas are covered. If a method is successful when no restrictions are made upon movement, then we know that the method could work in any circumstances where movement is unrestricted. In contrast to the other two strategies, the Random Walk method does not know when all of the search space has been covered, and as such there is no 'return to base' script. This leads to some duplicated effort, but has been deliberately left this way in order to see how unrestricted node movement would lead to a successful

As well as comparing these three, we also compare results from all three strategies where each UAR only exchanges data with the Command Station directly (i.e. no data is forwarded). The metrics discussed in Section 4.2.1 show that overall

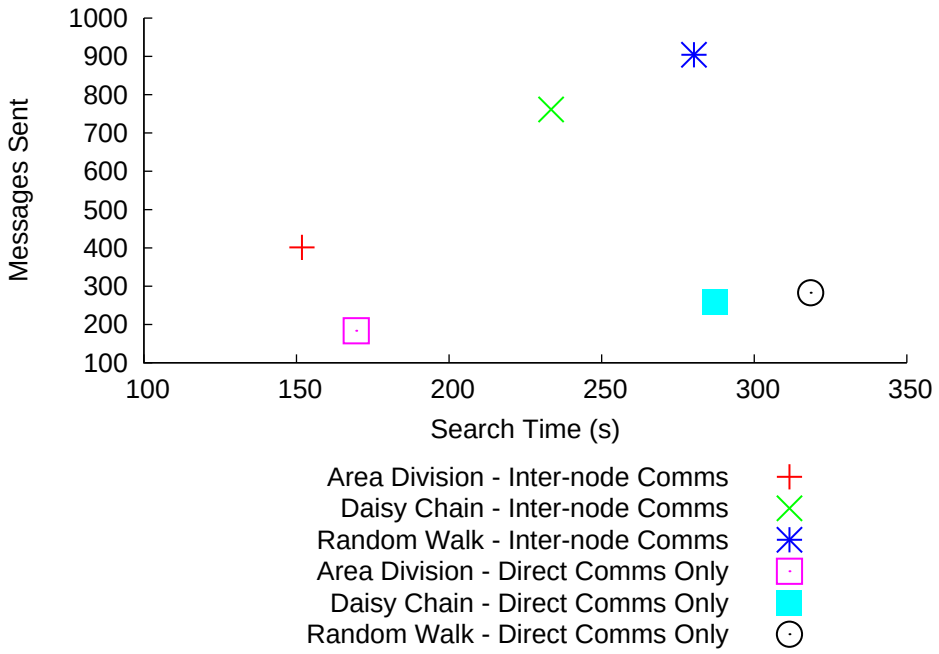


Figure 4.3: Messages sent against search time for a variety of search strategies.

search time and search completeness are the major issues in USAR. However, in our experiments we let the search continue until not only all targets had been found, but the coordinates of all those targets had been received at the Command Station. Therefore all experiments are complete, and the search time is defined as the time taken for all targets to be found and their coordinates received at the Command Station. Because we are focussing on the communication issues in USAR, we also measure the number of messages sent, the number of data items exchanged and the number of data items stored on each UAR. Results have been averaged over 40 runs.

Figure 4.3 shows how opportunistic inter-UAR communication can lower search times. Those marked 'Inter-node Comms' were able to pass data between any nodes, while those marked 'Direct Comms Only' were only able to pass data directly to the CS. The more effective strategies are those towards the bottom left of the chart, as they exhibit low search times and low overheads. Table 4.1 shows that opportunistic communication lowers search time by approximately 11-23% compared to communicating only with the CS, but only at the cost of large increase in data and messages being sent, which will take up a lot more bandwidth. To investigate scaling issues related to inter-UAR opportunistic communication, we repeated the experiment but varied the number of UARs from 3-12. Results show that additional UARs do not decrease search time consistently (Figure 4.4), except for the Daisy Chain method. This can be explained by the following:

	Search Time	*Data Sent*	*Messages Sent*
Area Division - Direct Comms Only	169.58	35	149
Area Division - Inter-node Comms	151.8	80	322
Difference	-11.7%	+ 129.5%	+ 116%
Daisy Chain - Direct Comms Only	286.99	58	201
Daisy Chain - Inter-node Comms	233.43	221	540
Difference	-22.9%	+ 282.3%	+ 168.7%
Random Walk - Direct Comms Only	318.5	69	214
Random Walk - Inter-node Comms	280.26	262	642
Difference	-13.7%	+ 281.8%	+ 199.7%

Table 4.1: Effects of opportunistic inter-UAR communication on Daisy Chain, Area Division and Random Walk for four UARs

adjusting the number of nodes in Area Division means altering the way that the area is divided. This can result in differences in efficiency, and therefore of speed with which the environment is searched. Also, the addition of extra robots will lead to interference; as the environment is now more cluttered, robots must spend more time navigating around each other. In terms of performing the search task, tasking additional UARs does not necessarily result in a faster search - the Area Division technique is faster with three UARs than eight UARs in a Daisy Chain or twelve UARs on a Random Walk - however the downside is a vast increase in communication cost (see Figure 4.5, which shows that the reduction in search time when deploying more UARs are small compared to the costs in bandwidth).

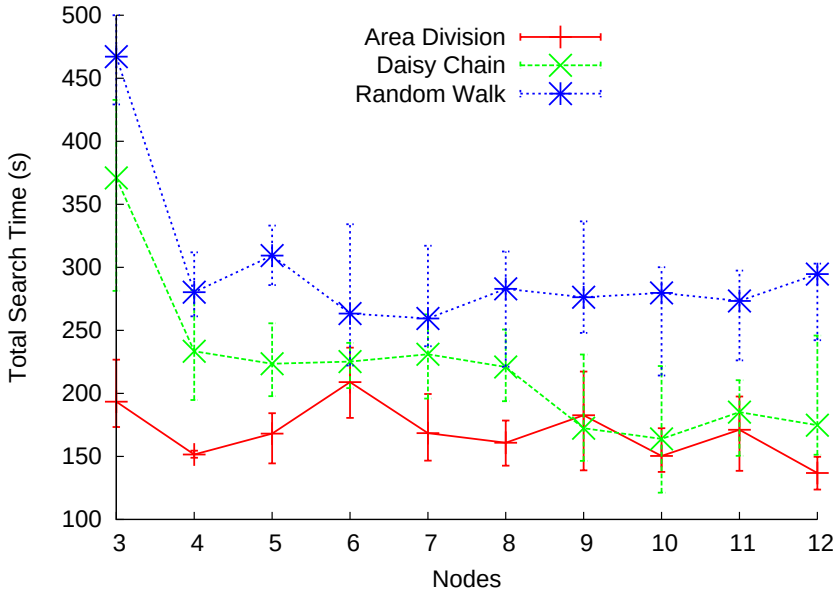


Figure 4.4: Search times for all three search methods for a number of UARs

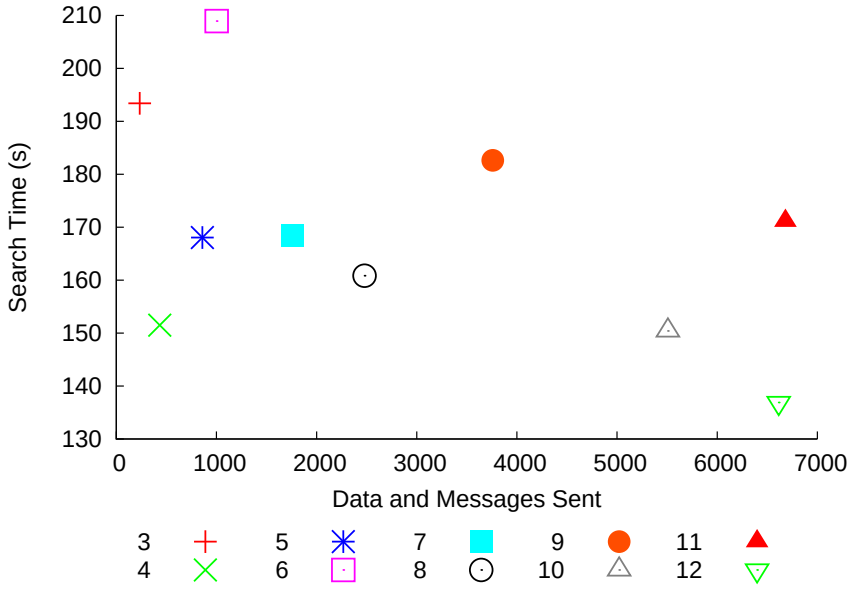


Figure 4.5: Search Time plotted against total transmissions for Area Division

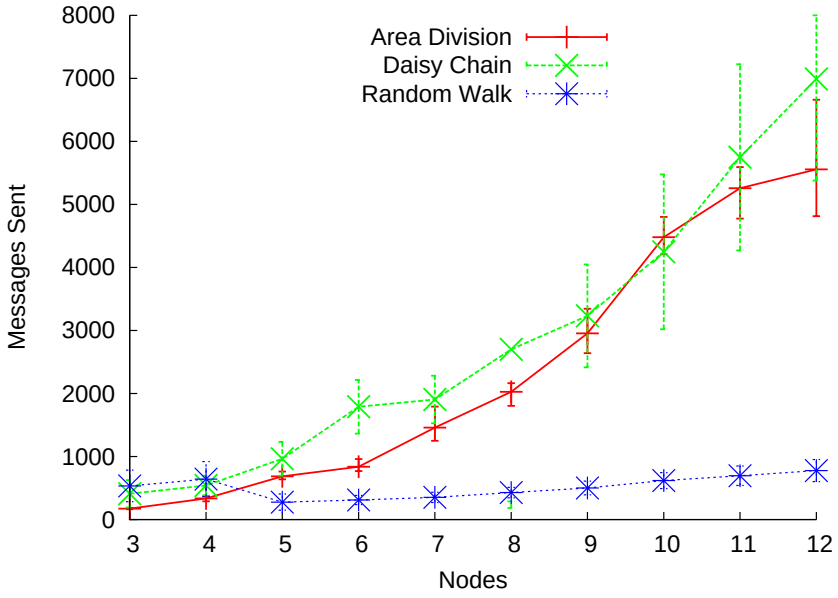


Figure 4.6: Messages sent for all three search methods for a number of UARs

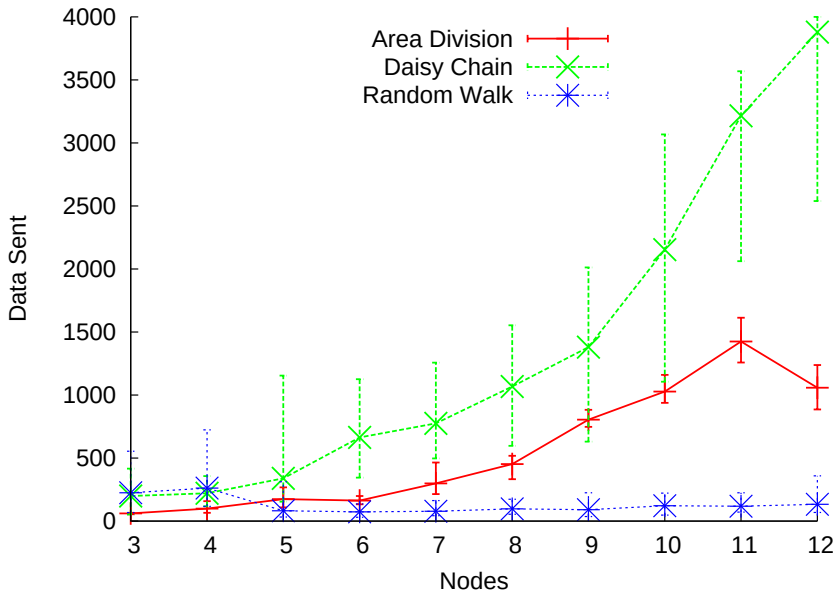


Figure 4.7: Data sent for all three search methods for a number of UARs

When tasking additional UARs there is a marked increase in both the number of messages sent (Figure 4.6) and the amount of data sent (Figure 4.7) for both Daisy Chain and Area Division. Figure 4.7 shows how Daisy Chain attracts the highest cost in bandwidth in return for its reduced search times. Results in Figures 4.6 and 4.7 show that Random Walk scales best out of the three methods; both Random Walk and Area Division scale much better than the Daisy Chain method. By analysing the number of networks joins (i.e. the number of times any two UARs came into communication range and performed handshakes) we see that the Random Walk method has far fewer joins than either of the other two methods, which means that the Random Walk technique creates a less well-connected network as it has fewest interactions as seen in Figure 4.8. By dividing the number of messages sent by the number of joins (Figure 4.9) we can see that all three techniques have a similar number of messages sent per interaction, as is to be expected. This allows us to conclude that the reason for the lower use of bandwidth in the Random Walk is because of the lack of connectivity. However, this also gives us some insight into the possibility of highly-connected UARs communicating too often; in our experiment handshakes only lasted until two UARs were out of range, but this could be altered to a time period or similar method in an effort to decrease the high use of bandwidth when the network has intermittent connectivity.

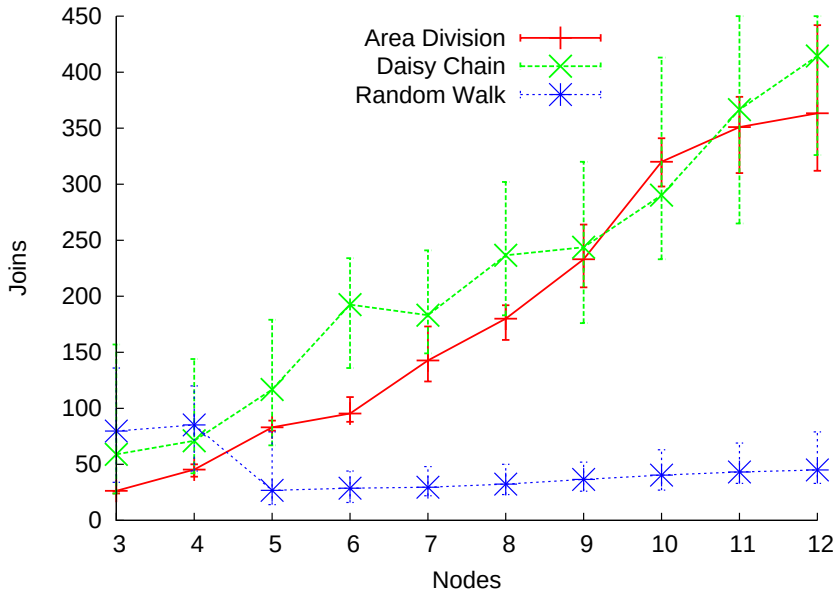


Figure 4.8: Number of joins for all three search methods for a number of UARs

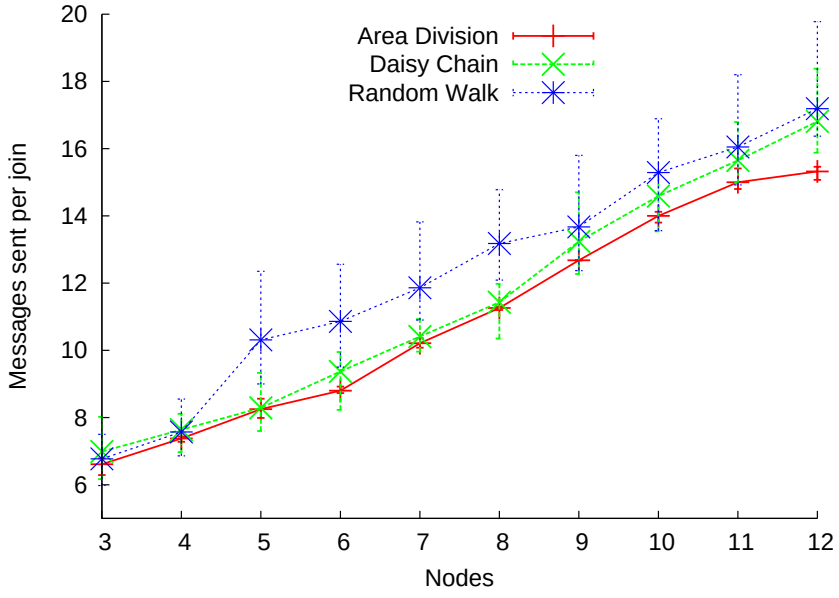


Figure 4.9: Messages sent per join for all three search methods for a number of UARs

4.3 Summary

Performing proof of concept experiments in simulation has shown that four UARs moving randomly and communicating opportunistically are capable of performing data relay without affecting the primary task of any UAR. Further experimentation compared three search techniques and found that dividing the search area between UARs while exchanging data opportunistically had the fastest search time while maintaining a reasonable bandwidth cost. We also show that the major limiting factor in bandwidth use was the number of interactions between UARs; our experiments have proved that exchanging data whenever robots come into range is too inefficient.

Opportunistic networks require intermittent connectivity to ensure that data propagates quickly. However, if there are too many interactions that do not exchange new data then the result can be a huge increase in the number of handshaking messages exchanged. Further experimentation is required to find a solution. The major limiting factor for the opportunistic relay is whether or not nodes get close enough to exchange data. When following a random walk or a thorough search pattern it is inevitable that data will eventually filter back to a Command Station. Random walk was the only technique that did not dictate any particular motion patterns; its success shows that opportunistic data exchange could potentially be used in a wide range of circumstances and still prove successful without making any movenment demands. However, in some scenarios UARs will have specific tasks to perform which could mean they never come within communications range of other nodes, meaning that some data is unable to filter back to the Command Station. Further work is required to recreate some behaviours typical of UARs or humans deployed in USAR, to investigate the effects of more realistic movement on the ability to relay data effectively, and how to overcome gaps in the communication field.

The final area that needs to be addressed is to ensure that whenever data are exchanged, the transfer is completed efficiently and effectively. Nodes in an opportunistic network must make the most of each and every opportunity to exchange data; when meeting anohter node they should both end up with complete copies of all data stored on either node. In order to preserve battery life - and to ensure that transmissions can occur within a small window - this process must be as fast and efficient as possible. It is with this in mind that we turn attention to look at data propagation for mobile ad hoc networks in Chapter 5.

Chapter 5

Data Exchange Mechanism - DEM

5.1 Introduction

In Chapter 4, we saw how an opportunistic network can be deployed in a USAR environment. Results showed that a high number of interactions led to a significant overhead in terms of exchanging data between nodes, where a node could be either a UAR or a wearable computer attached to either a human rescue worker or a rescue dog. Reducing overheads could lead to fewer replicated data and fewer messages being exchanged, and therefore a reduction in resource consumption. This could lead directly to energy, storage and bandwidth savings, reduced noise in terms of signal interference, and fewer lost packets. This would, in turn, lead to a corresponding increase in deployment life for each UAR. This would allow each UAR to operate in the field for a longer duration, increasing the number of available UARs, thereby speeding up tasks and potentially helping to save lives. In this chapter we will investigate data propagation methods for USAR.

5.2 Background

In mobile ad hoc networks (MANETs), mobile nodes form small networks whenever two or more nodes come within communications range. MANETs are used in applications where whole-network connectivity is unavailable; examples include UAVS acting as long-range scouts for the military [23], search robots operating in bunkers [62], or wildlife tracking [44]. The lack of network-wide connectivity could be due to a large geographic area, lack of technology such as antennas, intervening structures that prevent transmission or cause interference, loss of power, or saturation of existing links. In typical MANET applications, nodes are distributed

in a 'data collection area' where they must perform their task. Upon mission completion, termination, or simply whenever data are required, data must be relayed back to some fixed position processing facility, although sometimes the collection method uses mobility such as using a vehicle to pass by the nodes [44]. When data is forced to wait for a connection to appear like this, yet the system is able to cope with these propagation delays, the network is termed *delay tolerant*.

Without reliable long distance communication, mobile nodes must resort to using node mobility in conjunction with short range communications in order that data can propagate. The most simple propagation mechanism is a flooding approach, where any newly sensed or received data are sent on to all immediately connected neighbours greedily. This ensures that all data are received by all UARs with maximum theoretical speed [5], although there is the risk of overflowing the system with data, and this only works if there is sufficient available bandwidth. MANETS are generally battery powered, and as such have a limited energy supply. This means that an efficient mechanism for exchanging data can save power and increase the runtime of each node. We looked at existing propagation mechanisms for MANETs in Section 2.6, and now turn out attention to our own scheme for USAR scenarios.

5.3 Opportunistic Data Update

As seen in Section 2.6, flooding is the fastest way for data to pass through a network consisting of mobile nodes, assuming that node movement is random and uncontrolled. However, flooding sends a lot of messages. Power consumption is a limiting factor in MANETs, so we aim to maintain the speed of flooding while minimising the number of data items in the system and the number of messages sent. Our solution is a mechanism that improves efficiency so that only those data items that are required to fully update both nodes are exchanged. Calculation of which data are required is based on the comparison of data counters, which is achieved via a handshake message, after which data are requested. Once each data item reaches its final destination, a receipt is introduced into the system. This can be used in a number of ways: firstly, any receipted data does not need to be exchanged between nodes in the field, unless required for world models or mission data; if data are required for this purpose then receipted data can also be exchanged. Receipts can also be used for the purposes of data deletion; receipted data can be safely deleted to save storage space without fear of permanent loss, since the UAR knows that the data has been received at the Command Station. We aim to provide a solution that does not rely on dedicated relay nodes, nor

dictates any particular movement patterns; as such it should be usable under any circumstances where movement is unrestricted.

5.4 DEM

Each node in our system stores a triple corresponding to each other node; this triple is made up of a node ID, a receipt number, and a data counter.

- node ID: whenever a node adds some new data (not including data received from other nodes), it will append its node ID as the origin of that data; this allows all data to be tracked back to source, and acts as half of a source-destination pair. When data is received from another node, that data will be accompanied by the node ID of the originator.
- data counter: whenever a node adds some new data (not including data received from other nodes), it assigns each datum a number corresponding to its own data counter, which is then incremented. If a node has created 10 data, then its data counter will be 10, for example. When data are received from another node, the data counter corresponding to the originator of that data will be increased to match the highest received data counter for the originator.
- receipt number: receipts are introduced, in our scenarios, by the Command Station. Receipt numbers match a corresponding data counter; when datum from a node is received by the CS, a receipt is returned to that node, where the receipt number matches the data counter for the received data.

Here is an example of a [*nodeID, datactr, receiptnum*] triple: [*node0, 15, 4*]. This means that this node has already received data that originated with *node0*, although they may never have been in direct contact. The most recent datum received had a counter of 15, and the most recently received receipt number was 4.

5.4.1 Handshaking

In our simulation, our neighbour discovery mechanism used odometry to check whenever any nodes came within a pre-set communication range of 4m. When applying these techniques to real robots, we would require a method for neighbour discovery using beacons, which is a research topic in its own right and beyond the scope of this paper. If two nodes are found to be within communication range, then each checks their handshake list to see whether this is a new connection; connections are maintained until nodes leave communications range, which vastly

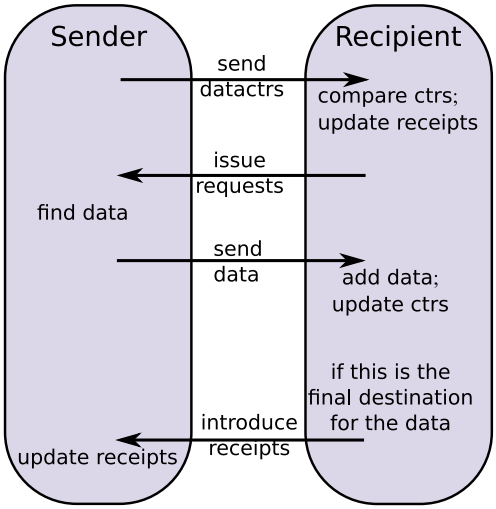


Figure 5.1: Diagrammatic explanation of the DEM

reduces the number of messages that are sent. If it is a new connection, then the node with the lower ID sends a handshake message and waits for a response; if successful this is called a *join* and data exchange can occur. Authentication can take place at this stage if required. Handshaking messages are sent individually to each participant, as DEM works only between pairs of nodes. If two nodes A & B are communicating and another node C comes within range of B but not A, A & C are not able to communicate directly. However, when B joins with C, if it receives any updates (either data or receipts) it will refresh its connection to A as well, ensuring that these updates can be immediately advertised to A. Note that in order to function correctly, DEM assumes that all node IDs are shared before the mission starts. If the mission commander is unsure whether all nodes will be committed, they should assume that nodes could be deployed. One area for future work is to alter DEM to allow nodes to be added or removed from the search effort.

Each node then sends a triple for every node in the system: [*nodeID, datactr, receiptnum*]. Each node compares the received triples to see which data items are required from the other, and then requests are sent, one request per node in the system. In order to calculate which data are required, firstly the recipient updates all receipt numbers; if any received receipt numbers are larger than the ones currently stored onboard, then the onboard ones are updated to match. Next, for each triple received (one per node), if the received data counter is larger than the current data counter for that node and is also larger than the receipt number for that node, then additional data is required to update this node. Request messages are then sent for all missing data.

5.4.2 Requests

A triple of [*nodeID,startCtr,endCtr*] is sent for each request. When a request is received, the appropriate data items are then sent. Once all requests are dealt with, if any received data has arrived at its final destination then a receipt can be introduced immediately. The data exchange mechanism is now complete; the method is shown in Figure 5.1. This leaves both nodes fully updated; after data exchange is complete, both parties should have identical values for the receipts and non-receipted data items. All data and receipts are cumulative, and *receiptnum* and *datactr* refer to the same data; hence only unreceipted data needs to be requested. This eliminates some replication of data. We impose no restrictions about the types of data that might be captured, which will depend heavily on the role of the node and the types of sensors it has on board. Instead we assume that all data can be stored as discrete items, and that each item is indexed using a simple counter.

Our method uses counters to allow for rapid comparison of data sets. An additional benefit of using counters with each data item is that it allows for post-mission analysis of the actions of each node. As an example, if the coordinates of the node are stored at regular intervals along with fuel levels, then this can be used to show the movements of the node during its mission. The same process can be used to show when (and approximately where) nodes exchanged data, allowing operators to build up a picture of what happened during the mission. For instance, if each join is logged as a data item with an appropriate counter by a node, then the corresponding data item can be found on the node that it joined with. The counters can be used to build up a picture of the movements of each node between joins, where they encountered hazards and from which direction. This could be particularly useful when debugging.

5.4.3 DEM Example

Let us imagine a snapshot of a system with three nodes. At the start of our snapshot, each node has the following stored as their [*nodeID, datactr, receiptnum*] triple. :

- Node0: [*node0, 15, 4*], [*node1, 9, 7*], [*CS, 0, 0*]
- Node1: [*node0, 6, 0*], [*node1, 12, 7*], [*CS, 0, 0*]
- CS: [*node0, 4, 4*], [*node1, 7, 7*], [*CS, 0, 0*]

Note that for simplicity, the Command Station (CS) has no original data; this is a fair assumption for our scenarios. Node 0 meets the CS, and each sends their triples. Each then compares the received *receiptnum* with their own; there are no differences, so no action is taken. Next, they compare *datactrs*; if any received counters are larger than both the onboard *datactr* and corresponding *receiptnum*, then some data is required and a request will be sent. The CS issues requests for data Node0:5-15 and data Node1:8-9. Node 0 receives these requests and sends the appropriate data. Once these data are received, CS sends receipt messages for all received data, and Node 0 updates its *receiptnum* accordingly. Node 0 does not require any data or receipts from CS, and so does not issue any requests. DEM is now complete, and these two now have the following triples:

- Node0: [*node0, 15, 15*], [*node1, 9, 9*], [*CS, 0, 0*]
- CS: [*node0, 15, 15*], [*node1, 9, 9*], [*CS, 0, 0*]

Later on, Node 0 makes contact with Node 1; we will assume that no new data has been added, so the triples at this point are:

- Node0: [*node0, 15, 15*], [*node1, 9, 9*], [*CS, 0, 0*]
- Node1: [*node0, 6, 0*], [*node1, 12, 7*], [*CS, 0, 0*]

After exchanging *datactrs*, Node 1 has its receipts updated; because Node 0 has no unreceipted data, Node 1 does not require any data and therefore issues no requests. Node 0 issues a request for data Node1:10-12; this request is processed by Node 1 and data is sent. Upon receipt at Node 0, *datactrs* are updated. The final triples are therefore:

- Node0: [*node0, 15, 15*], [*node1, 12, 9*], [*CS, 0, 0*]
- Node1: [*node0, 6, 15*], [*node1, 12, 9*], [*CS, 0, 0*]

Note the difference between the *datactr* and *receiptnum* held regarding Node0 differs on the two. This is because the *receiptnum* held at Node 0 was greater than the *datactr* held on Node1, so data items 7-15 were not required, having already been receipted by the CS.

5.5 Related Work

DEM is essentially a version of flooding. Roumeliotis and Bekey [75] implemented replace rules on the receiver side of their communication protocol. Vahdat and Becker [86] created the epidemic method which ensures that only required data is transferred by comparing data held on each node. Harras et al [34] propose that choosing to send data probabilistically can reduce overheads. Tower and Little [85] implemented a method called 'Active Curing' that compares ACK buffers for faster comparison of data between two nodes. These systems are covered in more detail in Section 2.6.2, along with an overview of other related work. Winfield [89] implemented an epidemic scheme on a group of mobile robots in simulation to test how data could spread as nodes moved around and formed ad hoc networks. His work is covered in greater detail in Section 2.8.1.

DEM differs from the epidemic approach in having a reduced overhead; DEM exchanges a single message per node in the system, each comprising a triple of integers, whereas sending a summary vector requires more messages as the index of every message is sent (Vahdat and Becker indicate that there are methods of compressing this data). Active Curing differs from DEM in several respects: firstly, Active Curing delays sending data forward in the system in order that acknowledgements can spread faster than data. This is not necessary in DEM because acknowledged data is not required, and because delaying data causes unnecessary propagation delay. Also, when data arrives at its target in DEM a receipt is sent immediately; this is not the case with ACK buffers in 'Active Curing', which are not exchanged until the next time that the final recipient meets another node. This means the node that exchanged data with this final recipient may continue to exchange receipted data with other nodes.

5.6 Method

A USAR scenario has been simulated using Player/Stage [33], where a team of autonomous nodes with no a priori knowledge of the environment move randomly by following an obstacle avoidance strategy. The motion of the nodes is deliberately random, since the aim of the experiment is to show that our opportunistic communication model will be effective at getting data from source to destination without dictating node movement. A series of eight target beacons were spread through the 16mx16m obstacle filled environment; four of the beacons represent victims, and four represent hazards. Additionally, at a regular interval each node

created an 'update' data item which contains its coordinates at that point in time. This means there are three different data types in the system; victim data, hazard data, and updates. A number of nodes were deployed, moving randomly to search the environment while another acts as a stationary 'Command Station'. Once the coordinates of all eight beacons were received at the Command Station, the experiment was complete.

In testing we distinguish between messages that do not contain data and those that do in order to assess the potential costs of transmitting large data items. We assume infinite buffer space, infinite storage, and that transfer windows are sufficiently large that all data can be transferred in time. However, DEM is robust to loss due to the unexpected termination of transfer, since all data exchanges are based on requests; unfulfilled requests will simply be made again.

Interference is an issue in these environments, but our algorithm does not take interference into account directly. DEM is robust to loss through its request-reply architecture, which ensures that if messages are missed or requests are not honoured, then they will simply be resent. We briefly investigated the issue of loss by randomly dropping received messages in another experiment that is not shown. Despite the fact that individual nodes may have been carrying outdated versions of data and not be fully updated, the result was still a fully functioning system, albeit one that was more likely to be inconsistent and not as efficient as it could have been without loss. One area for future work is to look at an effective method for detecting loss at the stage where triples are sent without requiring large numbers of acknowledgement messages. Note that other techniques [34, 85, 86, 89] do not explicitly deal with loss themselves.

Using a random walk search procedure, the following six different propagation mechanisms seen in Section 5.5 were implemented and compared:

- Flooding: forwards all data, with replace rules at the receiver side so that if any received data are already stored, they are dropped [75].
- Epidemic: performs a set comparison between a received summary vector and all onboard data. Requests are then issued for missing data. We have implemented a version that sends ACK messages immediately upon receipt of data [86].
- Probabilistic: As for Epidemic, except that requests for data are only returned with a 50% probability [34].

- Active Curing: this method uses counters which are compared in the same way as DEM. Data items are delayed while acknowledgements are not. In our implementation, data were delayed for a single join, so they would be first advertised at the second join following the data being added [85].
- DEM, without receipts; see Section 5.4 for more detail.
- DEM, with receipts; see Section 5.4 for more detail.

In order to analyse the results we must look at the differences in method for each of the six. Flooding uses replace rules on the receiver side, so it is we expect it to have the same amount of data held on any one node, but the amount of traffic to be higher than the other techniques. Epidemic's set comparison is computationally more expensive than the simple counter comparison used in Active Curing and DEM, and should require more messages to send a set summary of all onboard data. The Probabilistic method is the same, except that by only sending 50% of the requested data, we expect the number of data sent to be lower than Epidemic while the delay should be higher; we are not sure how this method will compare with DEM, other than expecting the Probabilistic method to have a larger delay to get data to the CS than DEM. DEM with the addition of receipts is expected to prove the cheapest method in terms of number of messages sent, while maintaining the speed of Epidemic or Flooding.

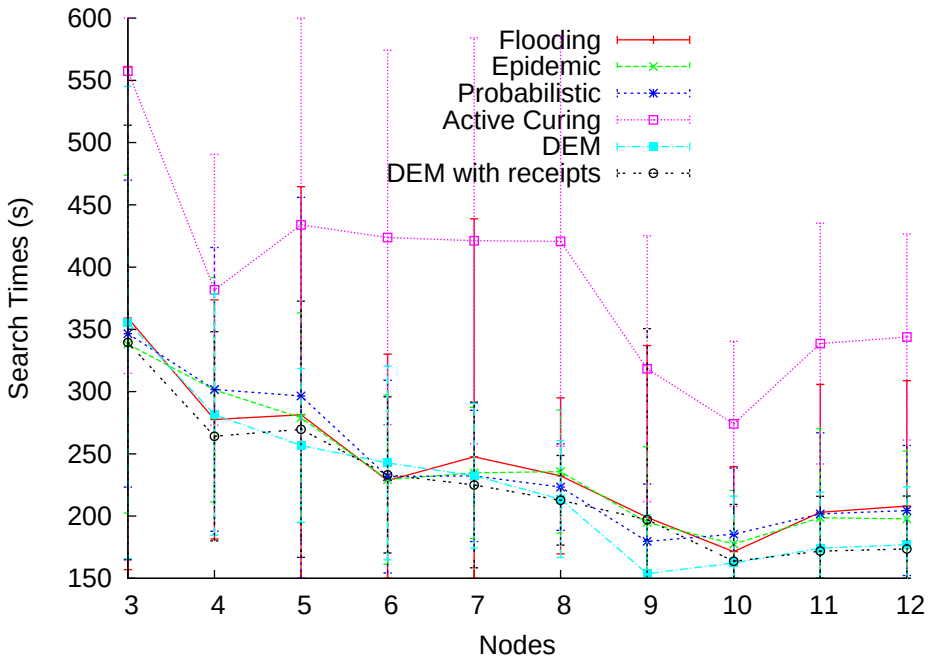


Figure 5.2: Comparison of delay times for propagation mechanisms

5.7 Results

Results, taken and averaged over forty runs per data point, show that five of the six methods have very similar search times for the experiment; see Figure 5.2. The exception was the Active Curing technique, which had a considerably larger delay than the others. This can be attributed to its technique of deliberately delaying the spread of unacknowledged data. The Probabilistic method did not have a larger delay than the others, something we incorrectly predicted; likewise it did not have any noticable difference in the number of data exchanged. One explanation for this could be that the nodes were generally connected to many other nodes, and so choosing whether or not to exchange data made little difference since at least one node would be chosen to receive all data. Crucially, DEM proved just as fast as the Epidemic method. All six methods resulted in a similar number of data items being generated, as seen in Figure 5.3, but not an equal number of data being sent, as seen in Figure 5.4; Flooding sent more data than all other methods - on average more than five times the number sent by DEM or Epidemic, although the variability of the amount sent during Flooding is also far higher than other methods, as seen by the relative sizes of error bars in Figure 5.4, which plot one standard deviation.

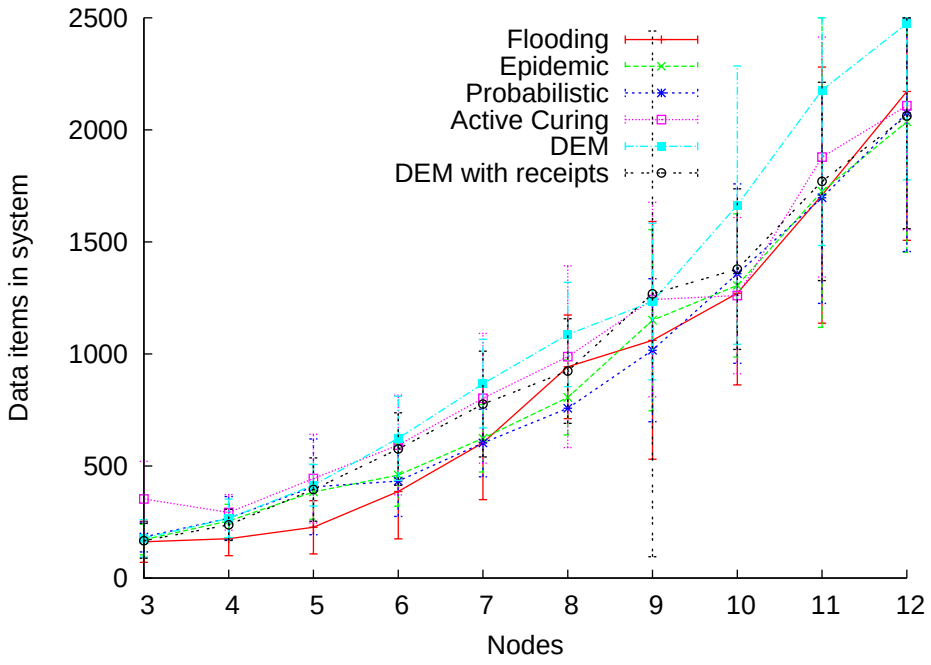


Figure 5.3: Comparison of data onboard for propagation mechanisms

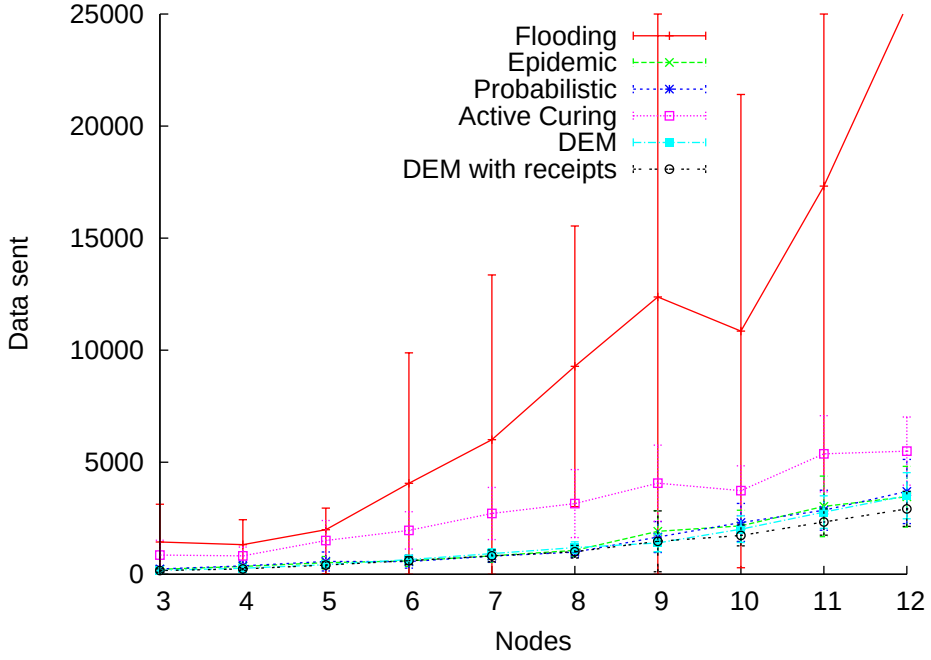


Figure 5.4: Comparison of data sent for propagation mechanisms

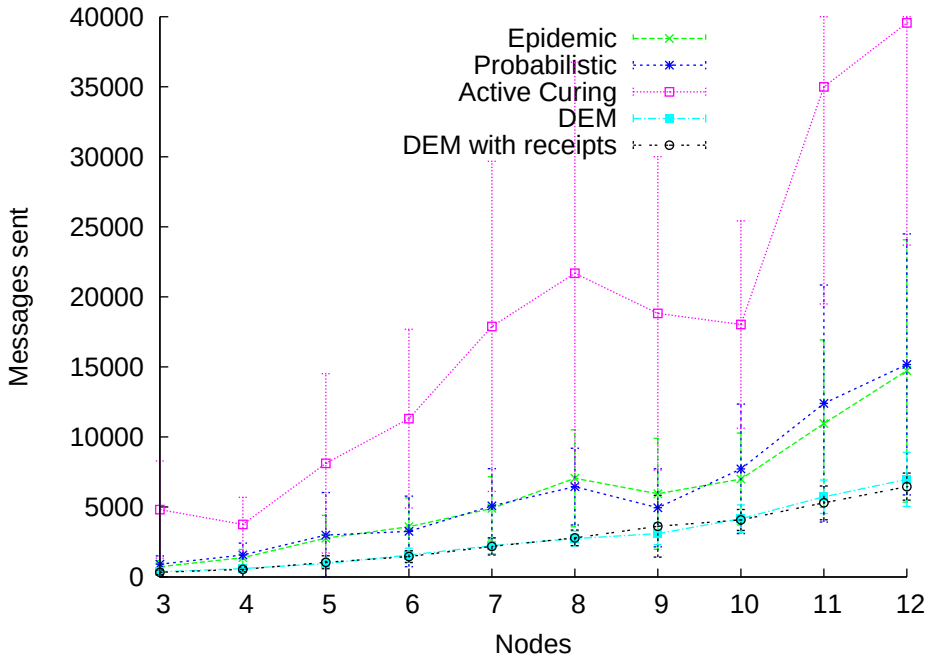


Figure 5.5: Comparison of message overhead for propagation mechanisms

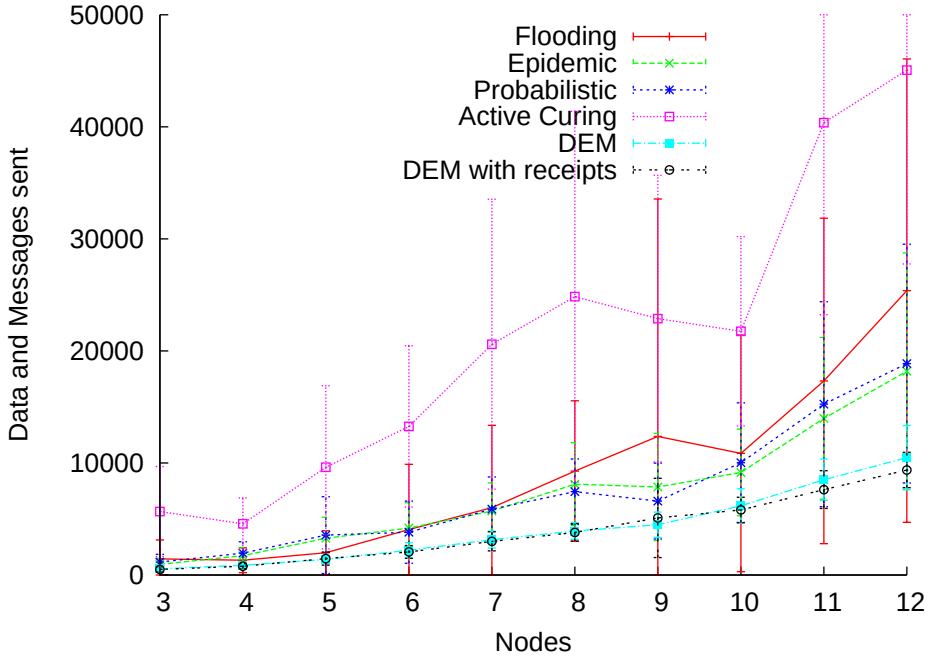


Figure 5.6: Comparison of data plus messages sent for each propagation mechanism

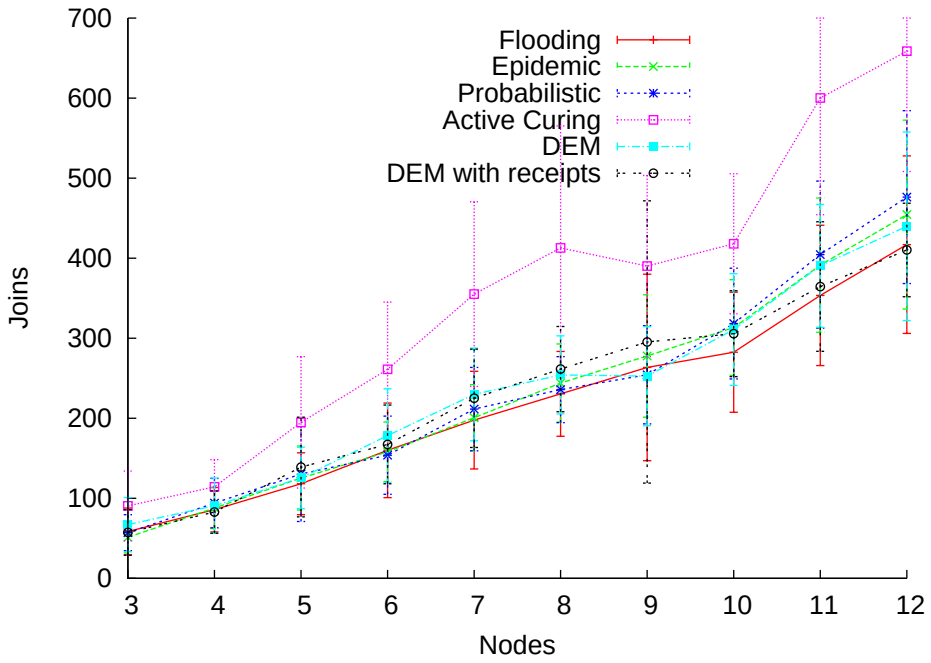


Figure 5.7: Comparison of number of joins for propagation mechanisms

There is a noticable increase in search times for all techniques for eleven nodes in Figure 5.2 following a distinct dip for ten nodes. This rise has a knock-on effect in other graphs where time is a factor; for instance, the total number of data or messages exchanged (Figures 5.4, 5.5 and 5.6), and the number of joins (Figure 5.7). The dip at ten nodes followed by a rise for eleven and twelve could be explained by concluding that ten nodes is the optimal number of nodes to have in this environment size; such a large number helps to create a better connected network, ensuring that propagation delay is minimised. However, deploying any more than ten nodes has a negative effect due in an increased level of inter-node interference; nodes frequently need to avoid each other, and therefore take longer to complete the search, something that outweights the benefits of having even more nodes forming the ad hoc network.

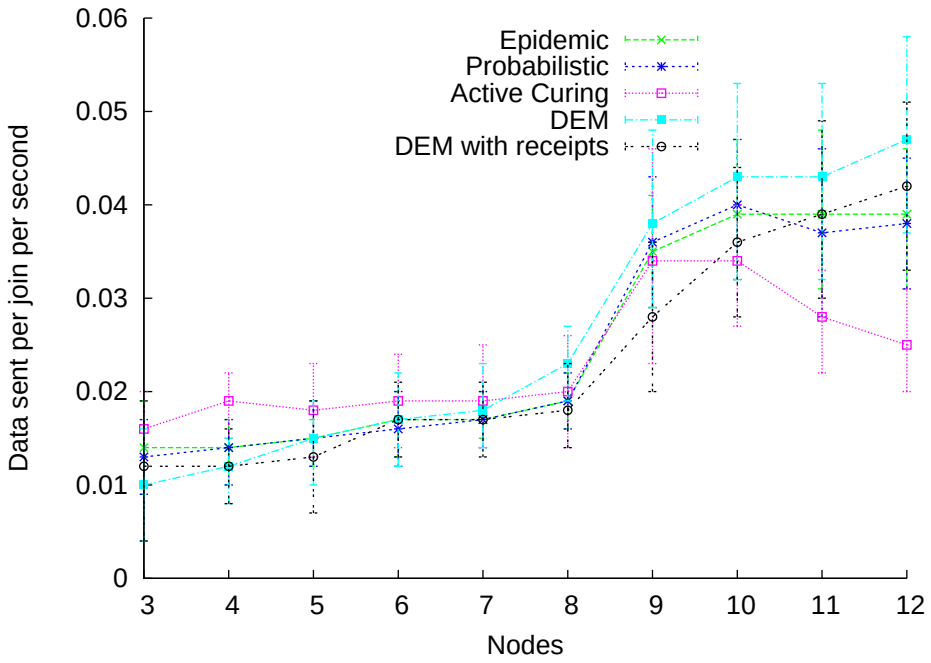


Figure 5.8: Average send rate for propagation mechanisms - Flooding omitted

Active Curing also had a noticably higher number than the other four methods, possibly due to the larger delay; in order to test this, we compared data sent per join per second in order to negate the differences caused by different numbers of joins or different lengths of delay. Results indicate that all methods are very similar except for Flooding, as shown in Figure 5.8; Flooding has been omitted from this diagram for clarity of the remaining results. It is worth noting that the Flooding method has no additional overhead in the form of handshaking messages that characterise the other methods. In Figure 5.8 it is possible to see a significant downturn for Active Curing for nodes=10-12, which can be explained by the significant corresponding increase in joins in Figure 5.7, as previously discussed. Note that Flooding has been omitted from this diagram due to scaling, and is shown in Figure 5.9; note that there is no downturn at nodes=10, since this figure shows the rate at which data is sent, therefore indicating that the hypothesis about the anomalies in Figure 5.2 is correct. We compared the number of messages required for each method in Figure 5.5. DEM - with or without receipts - uses on average half of the messages that the Epidemic or Probabilistic methods do, and approximately 16% of those required by Active Curing. Flooding has not been shown because it requires no messages.

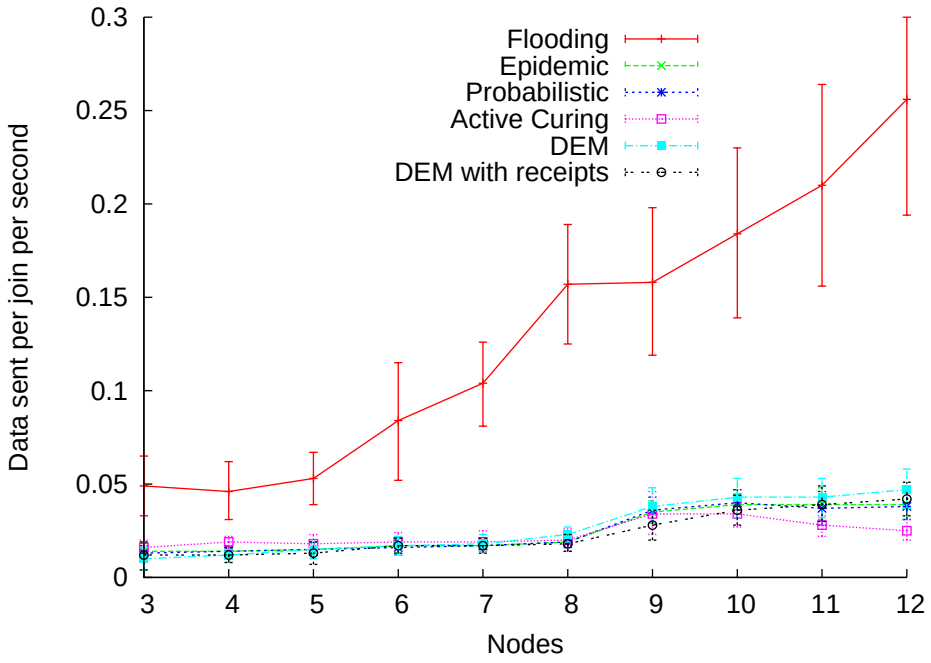


Figure 5.9: Comparison of average send rate for propagation mechanisms

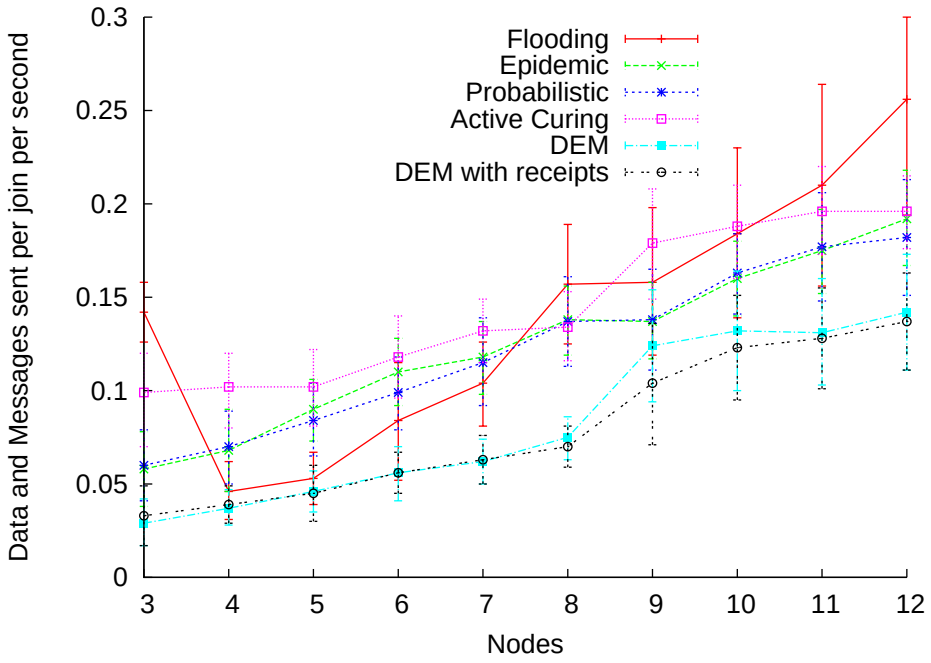


Figure 5.10: Comparison of data plus messages sent divided by joins per second for each propagation mechanism

	DEM	DEM with receipts
Flooding	51%	48%
Active Curing	18%	18%
Epidemic	55%	52%
Probabilistic	54%	51%

Table 5.1: Comparison of DEM methods against others for data plus messages sent

	DEM	DEM with receipts
Flooding	47%	45%
Active Curing	27%	26%
Epidemic	53%	51%
Probabilistic	53%	51%

Table 5.2: Comparison of DEM methods against others for data plus messages sent per join per second

Our comparisons assume zero loss; by analysing the amount of data stored between nodes (Figure 5.3), it will be possible to estimate memory requirements for each node. We also considered the worst case scenario where data are small enough that they can be treated the same as messages; results for the combined total of data sent and messages are shown in Figure 5.6. Active Curing proved highest; initially this was believed to be due to its larger delay, but by considering the number of data plus messages sent divided by joins per second in Figure 5.10, Active Curing was still found to be considerably more expensive than all other techniques. DEM with or without receipts is significantly cheaper than all other techniques, sending around half of the messages and data required by the other methods (Table 5.1 and 5.2). Figure 5.5 shows the message overhead in DEM is far lower than that used in the Epidemic or Probabilistic methods; Flooding has no overhead and hence is not shown. DEM also scales better than either. To ensure this was not due to our DEM experiments finishing earlier (and therefore having fewer chances to exchange data), or having more joins (and therefore a higher overhead), we took the total of data and messages and divided by time and joins; the result is seen in Figure 5.10, and shows that DEM consistently outperforms all other techniques. It is worth noticing that the addition of receipts in DEM still results in an overall reduction in the number of data exchanged, and that the improvement to DEM by adding receipts is a modest 5%. It is the use of counters to quickly compare data sets that has the larger saving.

5.8 Summary

DEM is a logical method for data exchange between two neighbours in a delay tolerant mobile ad hoc network. In simulation we showed that DEM maintains the speed of data propagation which characterises the Epidemic approach while reducing the number of exchanged messages by around 50%. Receipts can be used to further reduce the number of exchanged messages in DEM, but are not essential to its operation. DEM does not require any kind of time synchronisation, but if this is a requirement then local time can be stored alongside each data item. Our implementation assumed no losses; real wireless links are prone to loss due to a variety of reasons such as low-level interference and network breakages. While this might seem a large oversight, we are describing only a method for selecting which data items are transferred, rather than a transmission mechanism. As such, existing mechanisms such as TCP and its variants can be implemented to deal with the problems associated with loss. A discussion of these techniques is beyond the scope of this paper, but there are various implementations that attempt to overcome the problems associated with loss in wireless ad hoc networks; for a comprehensive survey, see Chen et al [18]. The goal of implementing DEM was to minimise energy consumption and unnecessary data exchange. While some mobile robots will be large enough that energy consumption isn't a major factor, DEM is designed to be used on any mobile ad hoc node; as such, it could be used on sensor networks or tiny mobile robots where power consumption is a vital metric.

Now that we have an effective mechanism for performing data exchange in an opportunistic network, we are able to implement DEM in a realistic USAR search, confident that the communication costs will be minimised. DEM has been designed so that it can be implemented using a number of technologies such as TCP, UDP or MAC protocols. Implementation of these remains an area for future work, as does using DEM on real robots or sensor networks.

Chapter 6

Trade-offs in USAR

6.1 Introduction

As described in Section 2.7, rescue workers or UARs performing a search in a USAR environment may be outside of communication range of the Command Station (CS), where data is collated and the rescue attempt is coordinated. Assuming that there is no global communication ability, upon discovering a victim, the rescuer will be unable to report their finding instantly. If human rescuers are deployed, they will often be in buddy pairs, and may be able to extract the victim at once. However, victims are often pinned and the rescuers might not be able to extract them; furthermore, the extraction of victims is extremely complex, requiring the knowledge of building engineers to ensure that moving any debris does not cause structural collapse. It is therefore often the case that victims are unable to be rescued immediately, and that the search process aims purely to locate all victims. UARs are currently unable to perform a rescue autonomously as described in Section 2.2, and are therefore limited to performing a search.

Given this scenario, upon finding a victim or hazard, the node has to make a decision about whether to continue searching or to return the newly found data to the CS. We developed several strategies in order to test which leads to the fastest search. We also looked at the deployment of one or two dedicated relay nodes, to see whether extending the communication range of the CS will help the overall search, or whether that UAR would have been better deployed by searching. We examine the tradeoff between reducing the search time and the delay in relaying data with the overall aim of reducing the time taken to get victim coordinates back to the CS when global communication is impossible.

There is a lack of information about how buildings are searched in real disasters; discussions with Technical Rescue officers working with the Leicestershire Fire Service [27] indicated that this is due to the wide variety of search spaces. It is therefore left to the senior officers to make decisions based on their personal experience. Typically, a building might be divided into floors, with rescue workers splitting up to search left and right down corridors, while using a communication channel running up the stairway. It is with these loose conditions in mind that our experiment has been designed. One field that has looked at the search method in depth is that of animal foraging in ecology, resulting in some mathematical models that can produce near-optimal behaviours in searching for food items. By applying one of these theories, called the Marginal Value Theorem (MVT) [8], to the USAR task, it may be able to produce a more effective search. We now investigate the application of MVT in Section 6.2.

6.2 Marginal Value Theorem

The Marginal Value Theorem (MVT; see Section 2.7.1 for further information) is an optimal foraging model from ecology that states that an animal should leave a patch once the rate at which food items can be found becomes sub-optimal. By plotting the cumulative gain over time, a gain curve can be generated, and the optimal patch residence time found.

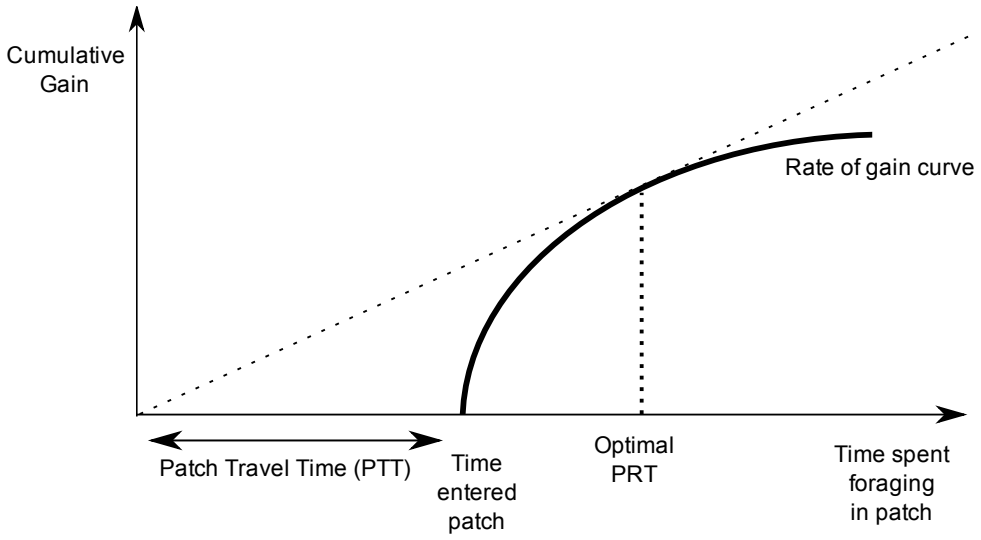


Figure 6.1: Explanation of how the optimal Patch Residence Time (PRT) is calculated

MVT has been applied previously to mobile robotics by Andrews et al [1], where vehicles (animals) need to decide which tasks to perform (food items to process) in order to optimise energy gain. By analysing the energy gain over time, the optimal patch residency time was generated. Heuristics were developed to allow each robot to estimate its current rate of gain; if this drops below a critical threshold, then the robot can decide to leave the patch and search elsewhere. Andrews et al examined the decision that would be taken by the robot upon finding a task; robots could decide whether to process the task or to leave it, depending on the cost of processing the task and the expected gain from processing it. This is analagous to deciding whether individual victims are too badly injured or trapped to be moved, or whether they can be recovered quickly. However, as autonomous victim recovery is beyond the current technological scope (as seen in Section 2.2), in our scenarios all victims are treated the same. Because of this, there are no decisions to be made upon finding a victim, and hence direct application of Andrews et al's work is not applicable here. They make several assumptions that also make their work unsuitable for application in an USAR environment, such as knowing the expected rate of encounter with tasks a priori, having an infinite number of tasks to find (there are a limited number of victims in our scenarios), and using flying vehicles which are controlled by humans, rather than the autonomous land-based vehicles used in our scenario. They concluded that, since MVT uses average rate-of-gain to predict optimal patch residence times, any deviation from this average can lead to sub-optimal predictions in individual scenarios; therefore, if the standard deviation between patches is large, the overall model may be inaccurate. This work was extended by Pavlic et al [66] by altering the cost of movement, and the chance of detecting tasks, in relation to the vehicle's speed, which was altered in such a way as to maximise overall rate of reward. Dechaume-Moncharmont et al [25] looked at the trade-off between search and communication in social ants by analysing whether it is worth getting help in dealing with extremely profitable patches. Their study shows that waiting for assistance and attempting to share information can be counter-productive, and that it is often more beneficial to forage alone.

In real animals, there are also many other factors that must be taken into account, such as the time of day, amount of cover from predators, and time spent away from the creature's home, which could leave young animals vulnerable; animals must seek to maximise their energy gain while reducing mortality rates [37]. The maximum number of food items that can be carried is highly relevant, as carrying too much can slow the animal down and make it a target to predators. However, the large gains from such a prey item are often worth the rewards, and

may immediately be returned to a nest for storage [8]. Foraging animals may also have to make a choice between different foraging patches, each of which may have different prey items and availability rates, and therefore reward rates; animals are reluctant to leave a patch when other nearby patches are low on food [15]. Likewise, many different types of prey may be encountered en route. This choice may be affected by the time of year (which food types are abundant), as well as previous foraging trips which may have exhausted the patch, and whether the animal is a specialised or generalist feeder. These considerations have been left aside for this experiment, but may be used to make more realistic environments where predation is modelled through destruction or disablement of the UAR, and where decisions about whether to continue foraging or to return data can be make in real-time onboard the UAR. The method employed during the search varies from animal to animal, and may involve switching between strategies according to how much food has been found. In our scenario, the amount of data that can be captured about victims may also affect whether to return to the CS, but carrying data will not cause physical limitations in the ability of the UAR to move, like it might for a foraging animal.

6.3 Experimental Design

In light of related work discussed above, we have made the following assumptions:

- the movement patterns of each UAR are unknown to other UARs; there are no planned rendezvous between UARs. Hence any inter-UAR communication can be termed as opportunistic. In a USAR scenario, UAR movement would not be random, but it might be impossible to know precisely where any given UAR would be at a certain time since the environment is unknown a priori. In particular, it might not be possible to know how long it will take to search any given room. Therefore, attempting to rely on planning [92], world models [7] or historical data [44, 84] in order to route messages is a risky strategy. There are no planned rendezvous between nodes for the purpose of relaying data, even when one or more dedicated relay nodes are deployed.

- there is no global communication network; this could be due to the loss of radio frequencies or inability to penetrate structures [14].

- odometry readings are used for UAR localisation; within Player/Stage, these are error-free. In a real UAR deployment, a more accurate system such as visual SLAM would be required to ensure UAR localisation and navigation was accurate.

- UARs communicate by line of sight and in any direction; obstacles such as walls can degrade or block signals, limiting the range of any one link [14, 61, 64, 62, 63].

- data transfer can take place successfully regardless of the transfer window; this removes the issue of loss to avoid it affecting results at this stage in testing. We assume that with having small data and the use of a mechanism such as TCP can ensure that data exchange takes place successfully.

- the search space is large enough, and has sufficient obstacles that total coverage of the entire search space is not possible by a communications network [6].

- victims are static, since conscious victims can typically extract themselves from the environment [22]. Optimal foraging theory works well when the prey are static [8], so this should result in an accurate application of MVT to the USAR mission.

- all victims are placed within rooms, with rooms treated as patches, and corridors treated as empty areas between patches. MVT works well when applied to discrete patches that are interlinked with 'dead ground' that is not foraged; as such, rooms in our scenario represent patches, and corridors act as the linking space between rooms. Victims have been mostly placed near to room boundaries in order to give different search times within the same room, with each room containing between zero and three victims. Using a truly random victim distribution for each experimental run would introduce variables that would make analysis of the results difficult; by using a fixed number of distributions, it is possible to analyse the data knowing precisely where the victims were and what affect their placement had on the results.

- the search process is time-critical; the faster the victims are found, the better their chance of survival [20]. This means streamlining both the search itself (to locate the victims) and the data relay (to get the victim's coordinates back to the Command Station) and minimising the overall time taken for each vicim.

- there is little information about the environment a priori. Real rescue workers try to extract as much information about the inside of a structure as possible before committing themselves, but are unlikely to know much beyond the basic layout [22, 27]. As such, we assume that there is some information about the number of rooms in the building and their approximate location, and that these rooms are divided between the UARs prior to the mission

	Environment		
	1	2	3
Size	30m x 14m	20m x 20m	20m x 16m
Rooms	7	16	5
Corridors	1	4	3
Rooms per corridor	7	4	1.25

Table 6.1: Environment Comparison

start. Effectively the UARs know where the doorways are; once inside a room they follow a wall following algorithm, similar to the pattern followed by a firefighter [22].

- the CS moves into position and remains static until all UARs have sent a message to confirm that they have finished their search. We assume that the CS is an extension of the manned Command Station; it could be connected by tether or a guaranteed wireless link. Therefore, any data that reaches the CS are assumed to be known to human rescue workers. Firefighters often use stairways to run a communications channel, so this is an obvious starting point for our experiments [22].

- in experiments where one or more dedicated relays are deployed, the placement of those relays is designed to maximise the amount of area that can be covered - via line-of-sight wireless communication - to the CS. As such, relays sometimes redeploy in order to maximise their effectiveness. This is done when a message is received at the CS stating that particular rooms have been searched. More details are found in the section for each environment.

- victim detection is instantaneous. In reality, readings from multiple sensors would be required to ascertain whether or not a victim had been found; any of heat, movement, colour, shape, clothing and/or CO_2 emissions [10] could be used to locate victims. USAR UARs may also be able to classify victims according to how difficult they may be to extract and how badly injured they are, and could potentially use this information to judge how to act when finding a victim. However, this is currently beyond the scope of our work.

- three environments are used for testing. A summary of the characteristics of these is seen in Table 6.1.

6.3.1 Environment 1

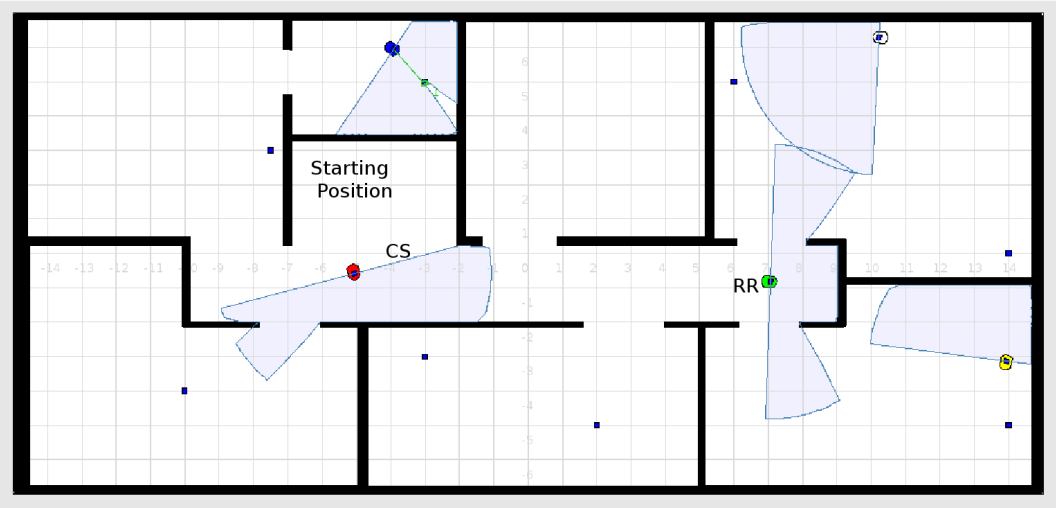


Figure 6.2: Environment 1 for our trade-off experiments

Environment 1 (see Figure 6.2) has a single corridor with rooms leading off it, representing the upstairs floor of a large house. The UARs begin their search at the marked area, representing a stairway. The position of the CS is indicated, and the UAR marked 'RR' acting as a dedicated relay. Six of the seven rooms are directly connected to the corridor, with one room connected via another room. UARs start in a space that represents a stairway; stairs are often used by firefighters to establish a communication link between floors, so this is a natural place for the CS in our scenario. Environment 1 measures 30m x 14m, and has seven rooms. When using a single relay, the relay is positioned at the far right of the corridor. When the right-most rooms have been searched, the relay relocates to get coverage of part of the interior of the middle rooms, starting with the bottom one, then the top. If a second relay is deployed, it is positioned at the far left of the corridor.

6.3.2 Environment 2

Environment 2 (see Figure 6.3) represents a simplified and smaller version of the second floor of the Haslegrave building at Loughborough University. The position of the CS is indicated. The starting position of the relay is marked with 'RR'. 14 rooms are situated on the outside of the building, with two further rooms and stairs in the interior block, which is surrounded by four corridors. The UARs are split into two groups: two UARs and the CS start at starting position 1, with the remaining two UARs at starting position 2. Environment 2 measures 20m x 20m, and has sixteen rooms. When a single dedicated relay is deployed, it is positioned at the top left corner of the corridors. When the left-hand rooms have

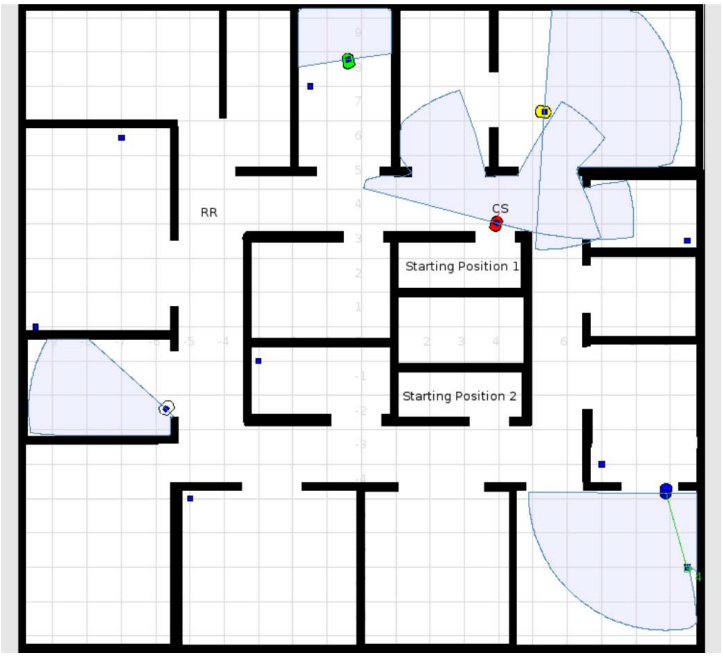


Figure 6.3: Environment 2 for our trade-off experiments

been searched, the relay relocates to the top-right of the environment in order to stay in communication with as many searching UARs as possible. If a second relay is deployed, it is positioned at the bottom left corner of the corridors.

6.3.3 Environment 3

Environment 3 (see Figure 6.4) is based on a simple office environment blueprint; the building has a single entry point leading to a long corridor which divides the floor plan, leading to a further two corridors. The position of the CS is indicated. The starting position of the relay is marked with 'RR'. Environment 3 measures 20m x 16m, and has five rooms. If a dedicated relay is deployed, in this environment remains static at the end of the top corridor. If a second relay is deployed, it is positioned near the centre of the map, at the junction between the central corridor and the top corridor to the right.

6.3.4 Measuring Success

There are many issues to consider when comparing methods, such as:

- How long does it take on average to find a victim? This is compared using mean search time (Mean ST).
- Once found, how long does it take to report a victim's coordinates back to the CS? This is compared using mean delay time (Mean DT).

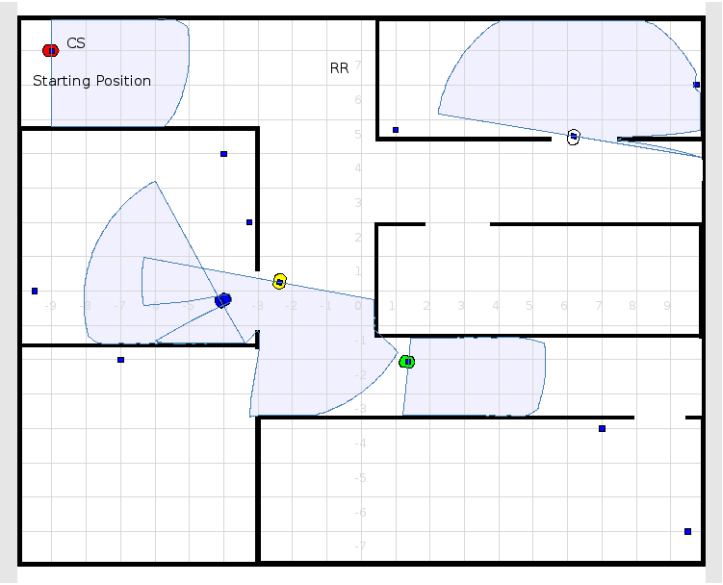


Figure 6.4: Environment 3 for our trade-off experiments

- What is the mean time taken from the start of the search to receiving a victim's data at the CS? This is compared using the mean total search time (Mean TST), which is the sum of search time and delay time.
- What time is the first victim discovered? (First TST).
- What percentage of victims are discovered? (Success Rate).
- Which technique has the highest rate of finding victims? This is compared by taking the number of victims multiplied by the Success Rate, which gives the mean number of victims found. The Mean TST is then divided by this figure to give the rate at which victims are discovered (Rate-of-Gain).
- Which method completes the entire search in the fastest time? This is compared by taking the largest mean TST from all victims (Earliest 100% finish).
- Is it better to minimise TST for the first few victims at the expense of those found last, or to discover all victims as quickly as possible? It is possible for two methods to have similar Mean TST and similar 100% finish times, yet discover victims at very different times. Likewise, two methods with identical TST could have different patterns; one that finds more victims than the other will be preferable, despite their identical TST. For each run of the experiment, the TST for each victim was recorded and ranked fastest to slowest; these were then averaged to give us a Mean TST for the first victim discovered through the eighth. These values are used to generate a

graph of cumulative victims found over time. This metric puts emphasis on minimising TST for each victim by taking the sum of the square of the mean TST for each of the victims (Sum of sq. means); squaring the mean TST will mean that a technique is penalised for having any victims TST very high.

- The cumulative gain (in terms of number of victims' coordinates received at the CS) over time can be modelled graphically as a gain curve, similar to those used to generate optimal PRT for MVT. The closer to the top left of the chart, the better the gain curve.

6.4 Experiment

Our experiment is a simulation built in Player/Stage [33], where four UARs perform a search of a number of different buildings where eight victims are distributed. Victims were placed arbitrarily in five different distributions, and were placed only in rooms, with each room having between 0-3 victims. Player/Stage has been chosen due to its abiliy to model the stochasicity inherent in any chaotic operation; minute differences in laser readings result in different paths being taken and different times as a result. Due to the almost infinite number of combinations of different environments to search, numbers of available UARs, number of victims, different search strategies and combination of those strategies employed across the UARs, any experiments performed will only be an indicator of which strategies could be effective. As such, it is not expected that any general theories will emerge as a result of this work, but rather that this work will act as a springboard for future research, both indicating which strategies should be pursued and which metrics and characteristics of the experiment are most important.

The UARs split the search space between them (see Section 6.3), aiming to enter rooms then adopt a wall-following algorithm until they exit the room, then head to another room and repeat the process until all rooms are searched. Rooms are allocated prior to the mission start; this replicates a situation where an incident commander has blueprints of the building and allocates each UAR a list of rooms to search. Whenever two UARs come within line-of-sight, in any direction, they exchange handshakes and issue requests for data they require, using DEM for the data transfer; full details on DEM can be found in Chapter 5. UARs exchange data about which victims have been discovered, and which rooms have been searched to avoid replicating tasks. If new data are discovered or received, new handshakes are sent to ensure that data relay can be performed immediately. Incomplete

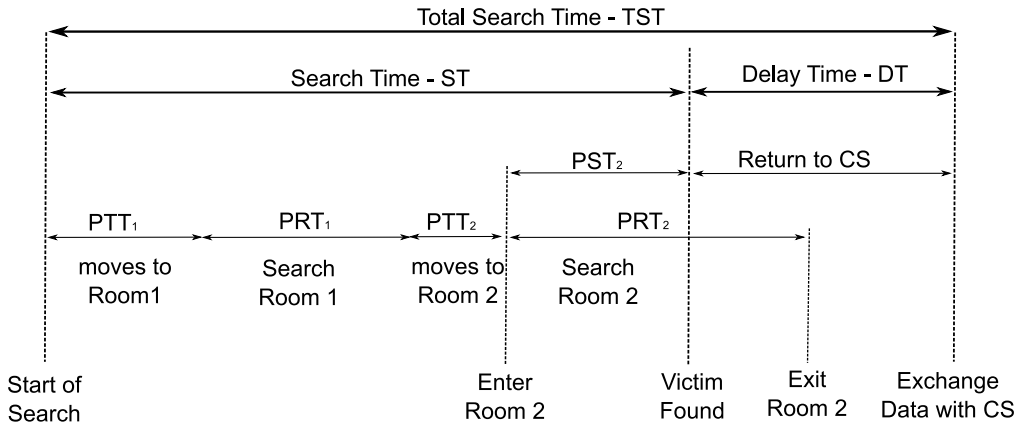


Figure 6.5: Breakdown of the terms used in Sections 2.7.1 and 6.4

runs - where UARs became trapped for instance - were discarded from the results to enable a clearer comparison of results. We use the following definitions for generating MVT data borrowed from Begon [8], which are shown in context in Figure 6.5 in the context of an example search conducted using Method 1:

- Patch Travel Time (PTT):- the time taken to travel to a particular room. Each room has an associated PTT for each run of the experiment.
- Patch Residence Time (PRT):- the total time that the UAR spent in a given room. The MVT process aims to generate an optimal PRT.
- Patch Search Time (PST):- the amount of time spent searching a room before finding a victim. If multiple victims are within a single room then they might have different PST but share the same PTT and PRT.

6.4.1 Hypotheses

Two hypotheses are being tested, based on the literature survey:

1. the application of a biological foraging strategy can lead to improved searching strategy in USAR environments
2. UARs are best tasked helping with the search effort, rather than being deployed for data relay; hence the application of an opportunistic network will be the optimal strategy

In order to test these hypotheses, several strategies will be compared, as detailed in Section 6.4.2.

6.4.2 Search Strategies

Six search strategies are compared. Each is a scripted search, without any ability to dynamically adjust to changing situations. This has been done in order to capture the merits of each strategy, so that it will be possible later to look at how to adjust strategies autonomously to minimise overall TST:

- Method 0 is an attempt to minimise ST; the environment is searched thoroughly first, then after each UAR has completed its search it returns to exchange data with the CS. It is expected that this method will have the lowest possible ST while ensuring the entire environment is searched, but the maximum possible DT.

- Method 1 is an attempt to minimise DT; whenever a victim is discovered, the UAR immediately relays its coordinates to the CS by driving to within communication range, then returns to the victim before continuing its search. It is expected that this method will have the lowest possible DT, but the maximum possible ST. In Section 4.2.2, this method was compared to a group of randomly moving UARs communicating opportunistically, and found to be faster in two-thirds of experiments.

- Method 2 is an implementation of MVT on a group of UARs performing an urban search and rescue task. UARs are substitutes for animals, rooms in the building are treated as patches, and victims are substituted for food items. In order to implement MVT in our USAR scenario, data is gathered during experiments using Method 0. By recording PTT, PRT and PST data from 375 runs of Method 0 (corresponding to 3000 victims), we have generated sufficient data to generate the cumulative rate-of-gain curve for each particular environment (see Section 6.2 for more details). These data are then used to generate an optimal PRT which is used in Method 2 (as seen in Figure 6.1), and a mean PRT that is used for Method 3. Curves will be generated using a logarithmic progression in order to get a smooth curve in an effort to remove the issues of deviations found by Andrews et al. A tangent to this curve will then be drawn; the x intercept of the two lines is the optimal PRT. This optimal PRT is then used as a cut-off point during searches using Method 2; UARs proceed as per Method 0, except that if the amount of time spent in a particular room exceeds the optimal PRT, then that room is abandoned and the search proceeds elsewhere. It is expected that this method will ensure that UARs only remain in rooms while they are 'profitable' in terms of finding victims; as such, our hypothesis is that

application of MVT will result in lower TST than Method 0, but with the possibility of missing some victims.

- Method 3 is identical to Method 2, except that instead of using the optimal PRT from Method 0, the mean PRT is used instead. This method has been implemented to give a comparison against the MVT using optimal PRT; the hypothesis is that this method will not compare as well as Method 2, showing that the optimal PRT generated by proper application of MVT is a better technique than simply restricting PRT to an arbitrary figure (in this case, mean PRT).

- Method 4 differs from Method 1 in that whenever a victim is discovered, the UAR finishes searching the room prior to returning to the CS. The hypothesis is that Method 4 will outperform Method 1 when there are multiple victims within the same room by reducing travel times. Note that, if the number of rooms to be searched is low enough that no UAR has to search more than one room, Method 4 is effectively the same as Method 0.

- Method 5 uses one UAR as a dedicated relay robot (RR) which remains in contact with the CS, effectively extending the range over which the CS can communicate. The remaining three UARs therefore have more rooms to search between them; these UARs use Method 0. RR may reposition as rooms are searched, still within range of CS but attempting to supply network coverage to those areas that are currently being searched in order to minimise DT; a description of how this is achieved can be found in Sections 6.3.1, 6.3.2 and 6.3.3. Our hypothesis is that the deployment of RR will extend communication range of the CS, and that this will lead to reduced DT, but that the loss of a search UAR will lead to increased TST overall. Hence, that four UARs performing a full search and communicating opportunistically will outperform a system that attempts to make use of dedicated communication relays.

- Method 6 is similar to Method 5, except two UARs are used as dedicated relays, with the remaining two UARs performing the search.

6.4.3 Testing

Individual behaviours and functions were tested by visual inspection of the Player/Stage simulation. Once behaviours were ready to run, each individual node's expected behaviour was tested individually (to ensure that it could navigate through the environment correctly, could find victims and exchange data as expected), then run in conjunction with others to ensure that there were no unexpected bugs. Without any formal specification, testing was limited to the use of validation of input and output of individual functions, and there are numerous routines in place to ensure that the UARs cannot get caught in loops.

Given the natural stochasticity in these experiments, with UARs interacting with one other, slight variance in laser readings and variance in timing of UARs coming into communication range, it was necessary to run each experiment a number of times, then to take a mean result. Therefore, each different method was tested 75 times on five different victim distributions, for a total of 375 completed runs per method, corresponding to 3000 victims.

6.5 Results

6.5.1 Environment 1

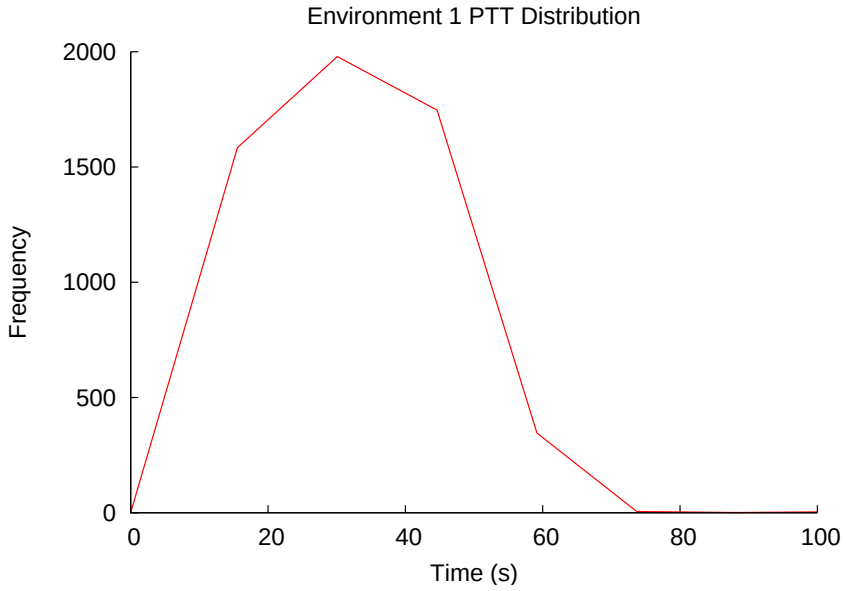


Figure 6.6: Distribution of Patch Travel Times for Environment 1

The results from Method 0 would determine the input criteria for Methods 2 and 3; the mean PTT of 22.69 seconds was found from Figure 6.6, and this was used to generate a curve in Figure 6.7; every PST for Method 0 has been plotted along the x axis, offset by the mean PTT. Each PST represents the time taken to find a victim once inside a room; these values are then plotted cumulatively along the y axis. The result is a rate-of-gain curve for the entire environment. By taking a tangent to this curve, the optimal PRT can be generated; in this case, $x = 46.34$ seconds.

Results in Figure 6.8 show the mean TST for each victim found by each method; these figures have been normalised to correspond to only found victims, and as such missing victims need to be taken into account. For the average victim, Method 1 clearly has the lowest DT, but at the cost of increased ST, while Method 0 has a low ST. Methods 0, 1, 4, 5 and 6 found 100% of victims. Method 2 used MVT to optimise the time spent searching each room; the result is a slightly reduced ST and DT than either Method 0 or 1, finding 99.1% of victims. Method 3 used the mean PRT from Method 0 as the maximum PRT, with the result that victims were often missed, as this method found just 69.5% of victims; the remaining 30% - which are those that take the longest to find - are essentially

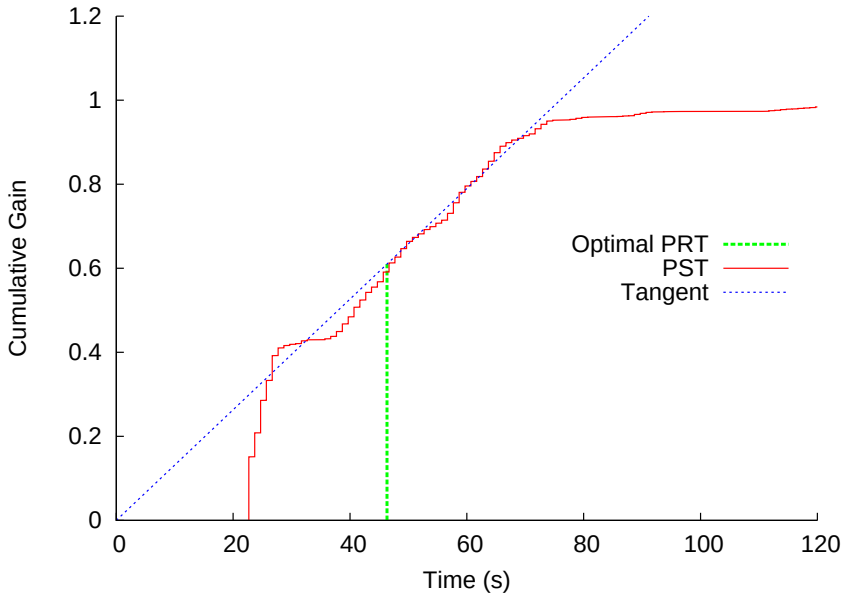


Figure 6.7: Rate-of-gain chart for PST in Environment 1

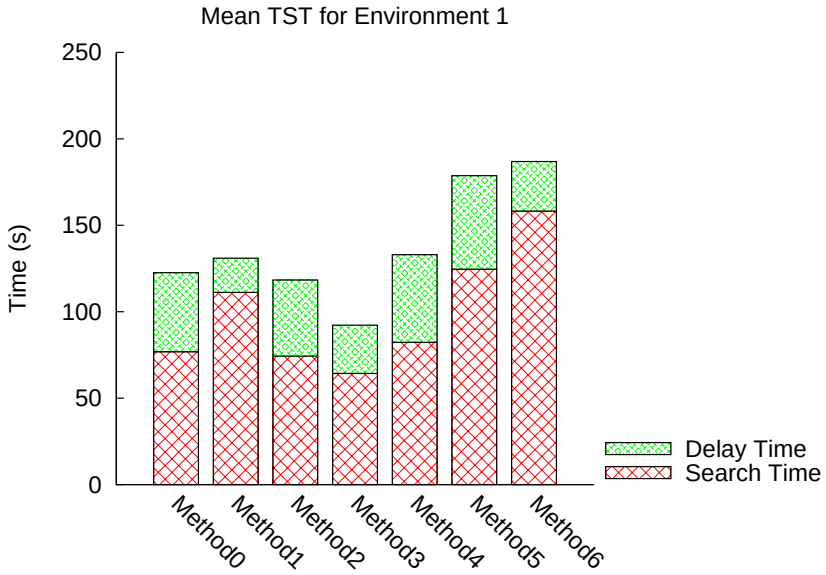


Figure 6.8: Chart showing the split between search time and delay time for each method in Environment 1

	Method						
	0	1	2	3	4	5	6
First TST	97	83	94	45	112	134	121
Mean TST	123	131	119	90	133	179	187
Success Rate	100%	100%	99%	70%	100%	100%	100%
Highest rate-of-gain	3.92	3.66	4.02	3.69	3.61	2.69	2.57
Earliest 100% finish	182	270	183	146	206	262	354
Sum of sq. means (k)	123	148	135	144	144	263	300

Table 6.2: Metrics - Results for Environment 1

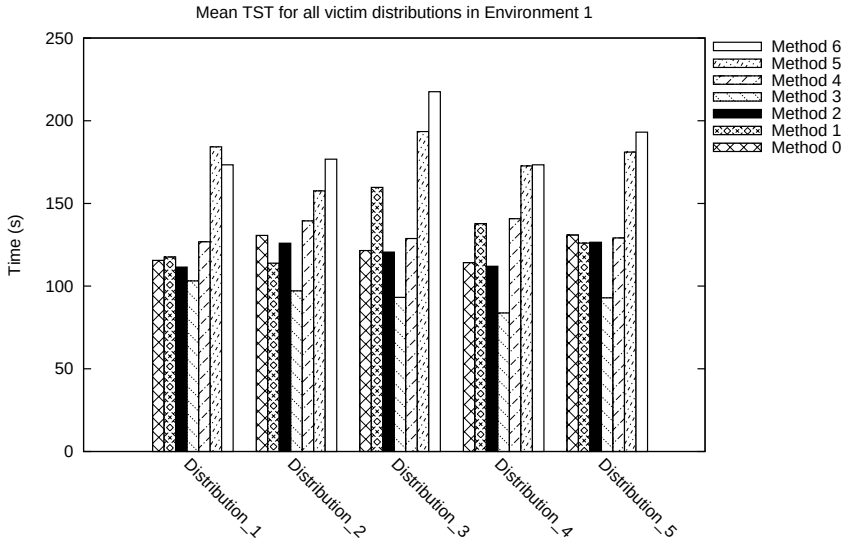


Figure 6.9: Mean Total Search Time for Environment 1

abandoned. Hence the low TST is somewhat misleading. Method 4 had increased DT and ST compared to Method 0; the choice of environment, being mostly rooms off a single corridor, may not be ideal for this method to show its benefits. Method 5 shows increased ST and DT, indicating that the relay UAR would have been better deployed to search in this scenario, with Method 6 performing worse still. From results (shown in Table 6.2), we can see that Method 2 has the highest rate-of-gain, finding and returning a mean of 4.02 victims per minute; Method 0 has the earliest 100% finish (excluding Method 3), taking on average 182 seconds to find and return all victims; and Method 2 has the lowest sum of squared means.

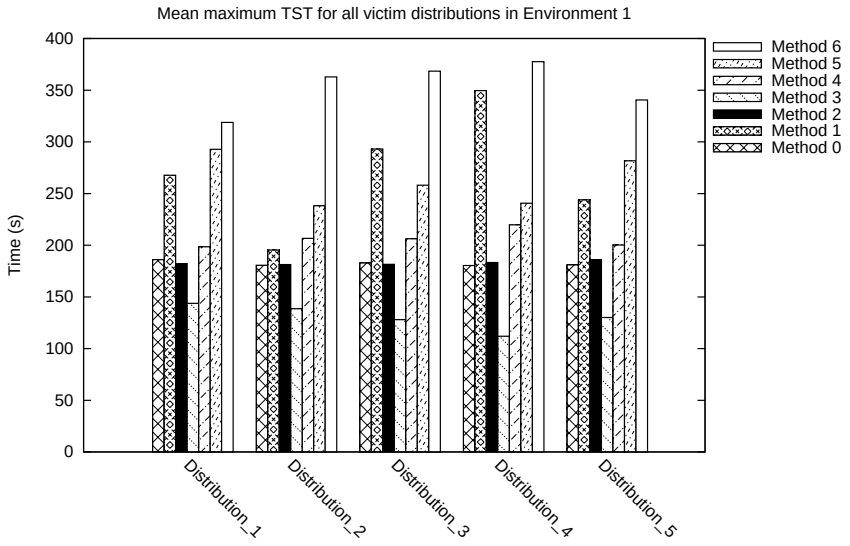


Figure 6.10: Mean maximum Total Search Time for Environment 1

Figure 6.10 shows the time taken to return the coordinates of all victims to the CS; while it appears initially that Method 3 is a clear winner, the success rate for this method is just 69.5% of victims found. With such a small number of victims being recovered with Method 3, it is unlikely that any SAR commander would choose to deploy this method. As such, in this environment, Method 3 can be discounted, revealing Methods 0 and 2 as having the lowest maximum TST. Figure 6.9 shows that the lowest average TST for any one victim varies between Methods 1 and 2; however, the reason for the low mean in Method 1 is because it returns the first few victims very quickly, at the expense of the final ones, as seen in Figure 6.11.

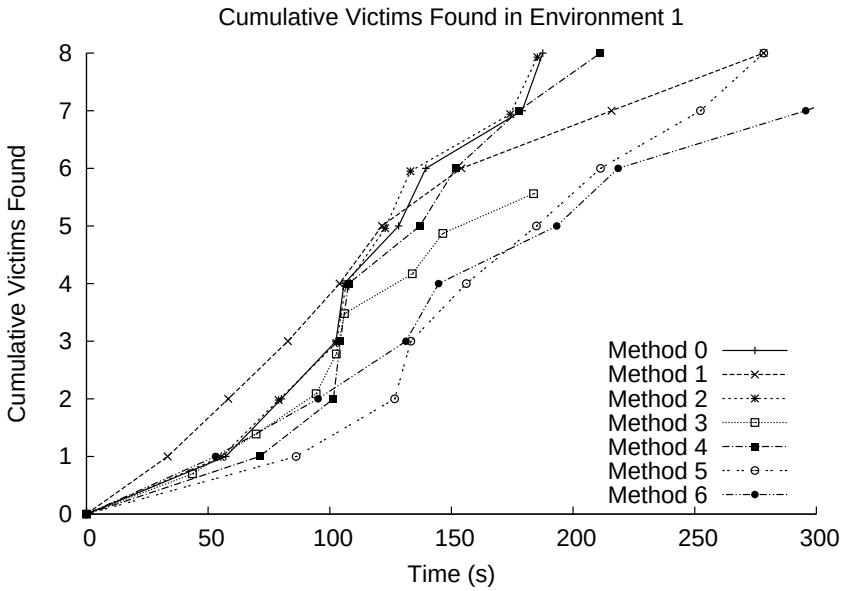
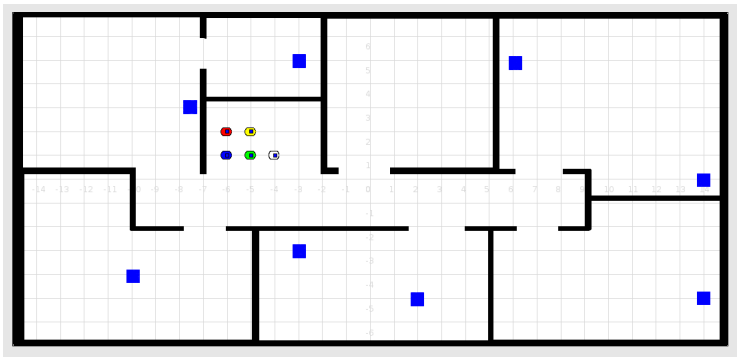


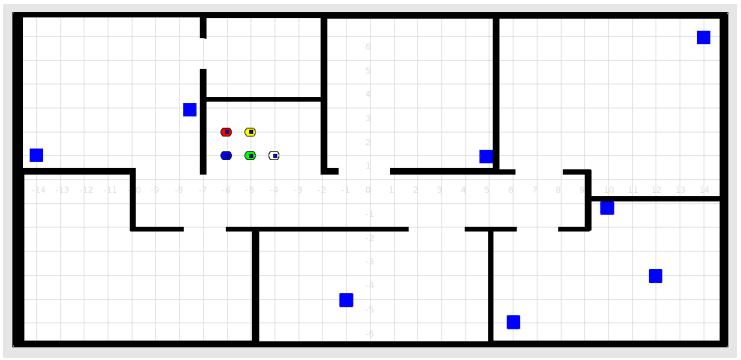
Figure 6.11: Cumulative gain curve for all methods in Environment 1.

Comparing the methods graphically in Figure 6.11 shows the distribution of times that victims are discovered for each method. Points plotted are the total time taken to find each victim and return that information to the CS. The chart has been scaled to show the majority of results with greater clarity. Method 3 has the earliest first return, while Method 1 tends to return the first five victims very quickly, at the expense of the remaining three which have a large TST. Method 2 is very similar to Method 0, with a slight improvement after the first five victims, justifying the application of MVT; however, for victim distributions 2, 4 and 5 (see Figure 6.12 for graphical representations of each distribution; victims have been enlarged for the sake of clarity in these diagrams), Method 0 performed better than Method 2. The time difference for earliest 100% finish between Methods 0 and 2 is no higher than 3%, indicating that there is no important difference between the two methods. Method 6 has by far the worst results in this environment.

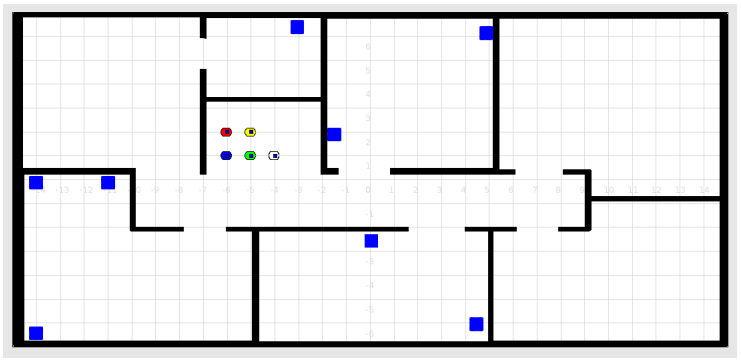
From the results we can conclude that Method 2 is both the most effective (highest rate-of-gain) for this environment, and is the most utilitarian method, doing the most good for the greatest number of people. However, the fact that Method 2 has no guarantee of finding all victims does count against it, despite its relatively high success rate of 99.1%; Method 0 has the earliest 100% finish with 188 seconds. Given that Method 2 is a special case of Method 0, the two performed relatively similarly, Method 2 showing a slight improvement in time at the cost of a very slight reduction in success rate. The poor performance of



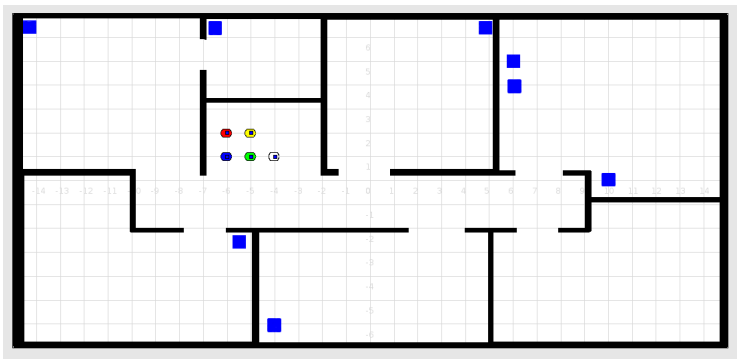
Victim Distribution 1



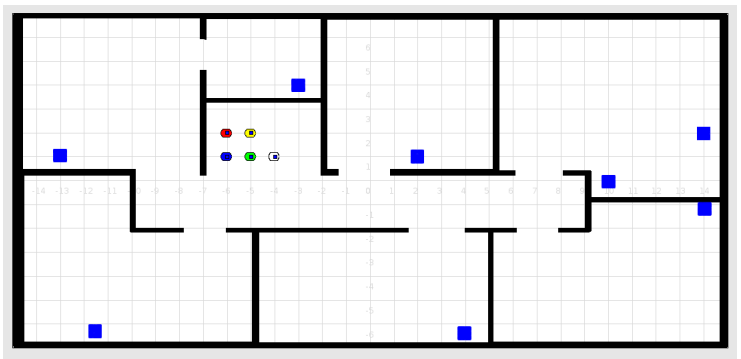
Victim Distribution 2



Victim Distribution 3



Victim Distribution 4



Victim Distribution 5

Figure 6.12: Victim distributions in Environment 1

	Distribution				
Victim	1	2	3	4	5
1	1	6	3	1	1
2	2	6	4	6	6
3	1	3	3	1	3
4	3	3	6	3	3
5	3	5	1	3	3
6	6	3	3	3	3
7	3	3	3	3	4
8	6	1	3	3	1

Table 6.3: Victim's choice of Method for Environment 1

Methods 5 and 6 may reflect that - in this environment - the CS has a relatively large communication span already, and the small increase afforded by the relay UAR(s) did not make up for the loss of a search UAR.

It is also important to look at the search from the point of view of each individual victim. If we were those victims, which method would we want to be deployed? By looking at the TST for individual victims across all Methods and victim distributions, the Method with the lowest TST was recorded in Table 6.3. In more than half of all experiments, any one victim would prefer that Method 3 be used because it would find them quickest of all methods. However, Method 3 finds less than 70% of victims, yet for those found, it is the best method.

6.5.2 Environment 2

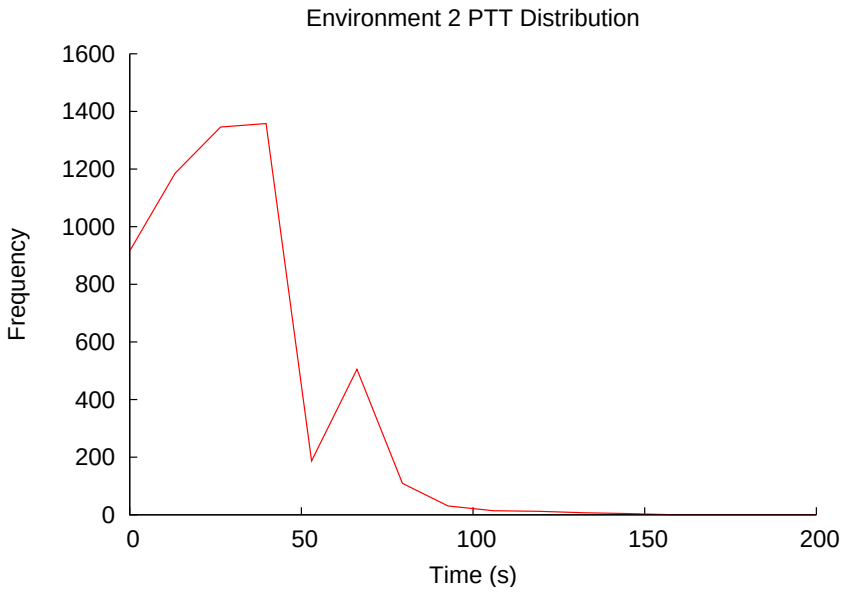


Figure 6.13: Distribution of Patch Travel Times for Environment 2

The results from Method 0 would determine the input criteria for Methods 2 and 3; the mean PTT of 22.72 seconds was found from Figure 6.13, and this was used to generate a curve in Figure 6.14, which gave an optimal PRT of 46.34 seconds. Results in Figure 6.15 show the mean TST for each victim found by each method. These figures have been normalised to take missing victims into account, but the success rates for Methods 2 and 3 were statistically close enough to 100% that this made little difference to the final graph; Method 2 found 2985 from 3000 victims (99.5%), Method 3 found 2984 from 3000 (99.47%). This is in contrast to Environment 1, and this can be explained by the smaller room sizes and larger PTT in comparison to Environment 1. Essentially, this shows us that, for Environment 2, the optimal strategy according to MVT is to exhaustively search each room. Therefore, Methods 0, 2 and 3 are almost identical.

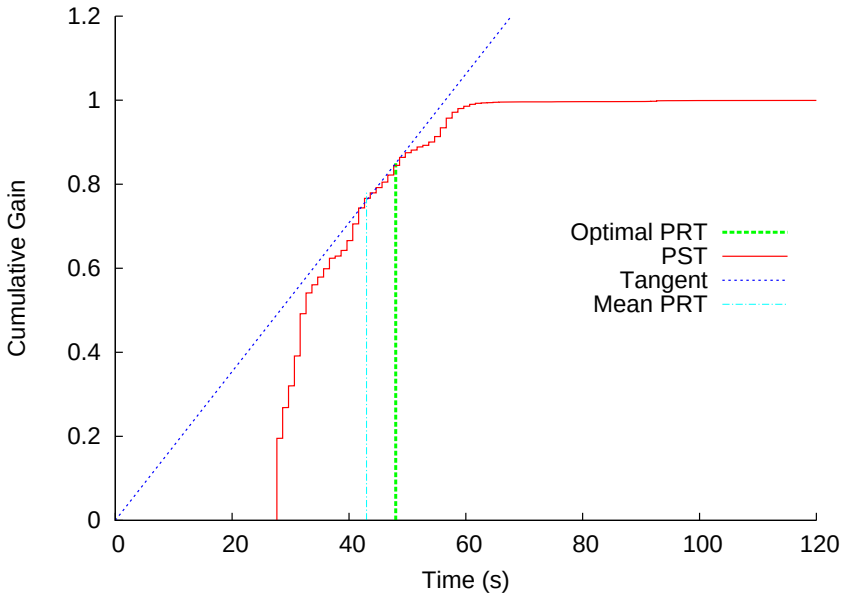


Figure 6.14: Rate-of-gain chart for PST in Environment 2

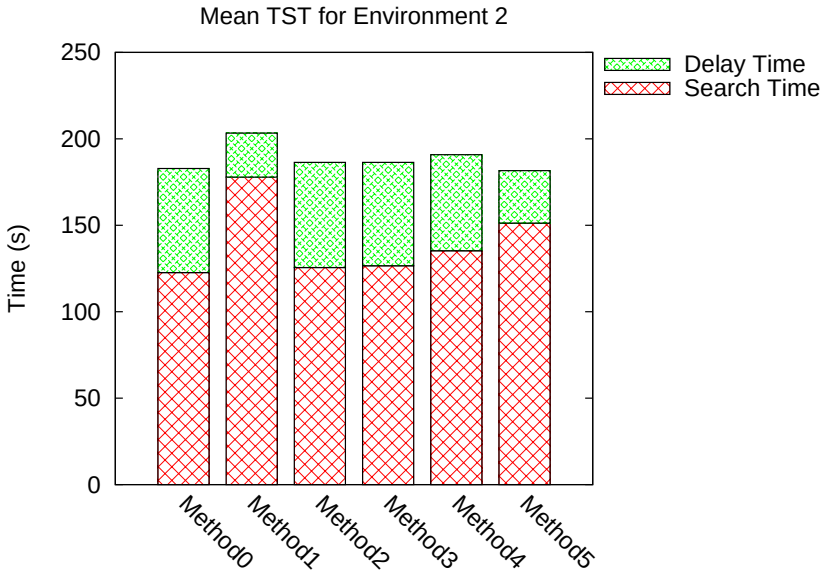
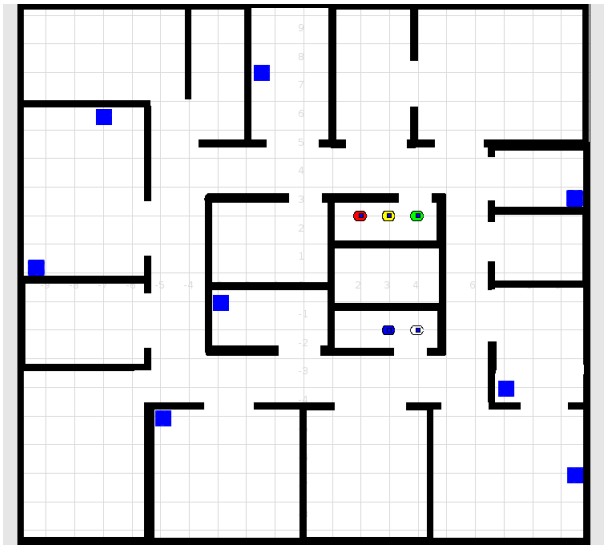
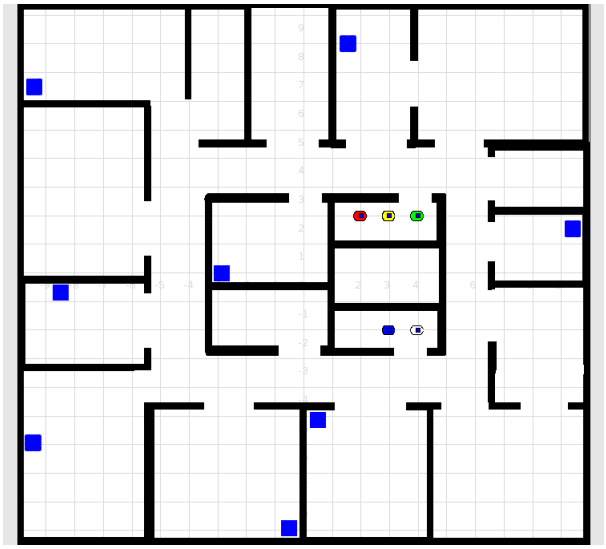


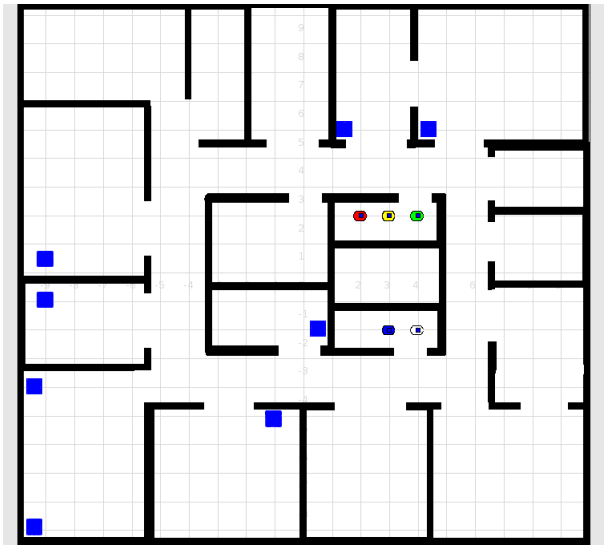
Figure 6.15: Chart showing the split between search time and delay time for each method in Environment 2



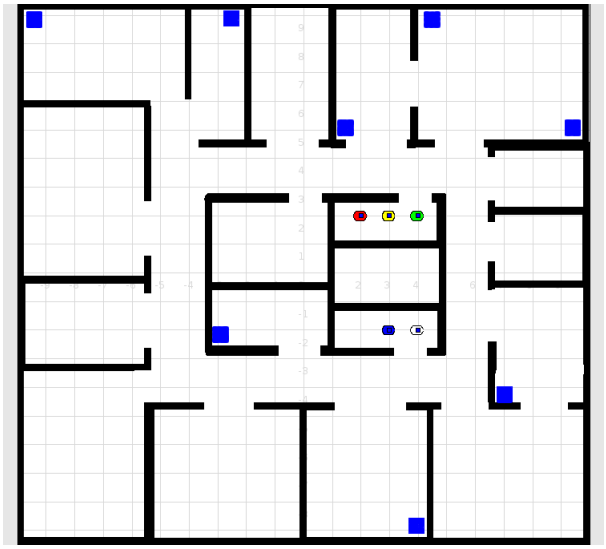
Victim Distribution 1



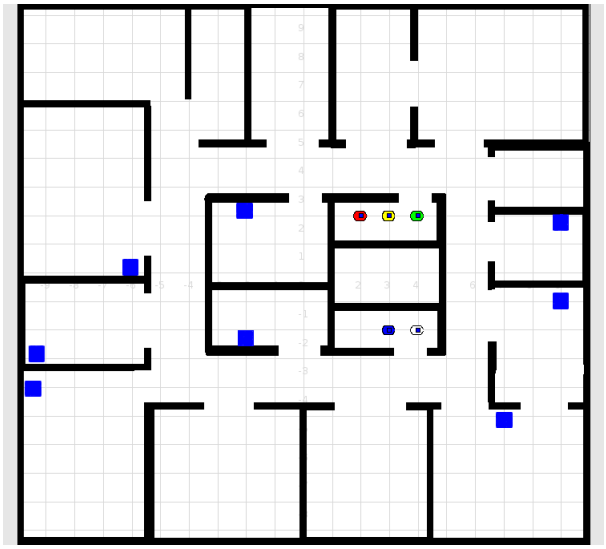
Victim Distribution 2



Victim Distribution 3



Victim Distribution 4



Victim Distribution 5

Figure 6.16: Victim distributions in Environment 2

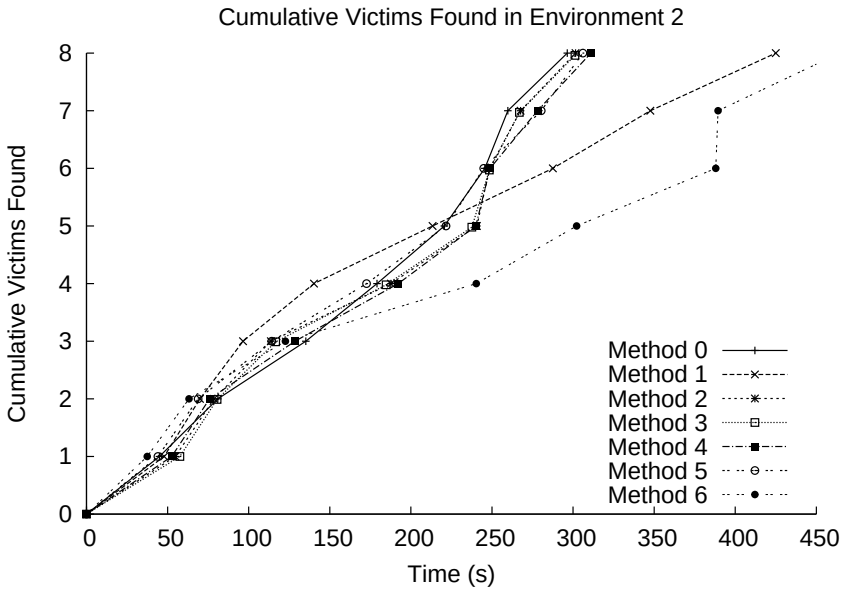


Figure 6.17: Cumulative gain curve for all methods in Environment 2

A breakdown of results by victim distributions (distributions shown in Figure 6.16, where victims have been enlarged for the sake of clarity, and results in Figures 6.18 and 6.19) showed that Method 4 proved to be the optimal technique for victim distribution 1, with a 10% time reduction compared to Method 0 and a 4% time reduction compared to Method 2. Futher examination showed that this result can be explained by the victim distribution being ideal for the application of Method 4; all of the victims were located either in the first room or final room that each UAR searched, or that the route to communicate with the CS was also in the same path as the next room that needed to be searched, and so no UAR needed to stray far from its path in order to communicate with the CS. These conditions meant that there were no victims that had a delayed ST due to any UAR returning to report victim coordinates to the CS, making it ideal for the application of Method 4. Method 5 performed best in distributions 2 and 5, with earliest return times taking as little as 71% of the time required by Method 0 in distribution 5; in this case, all victims were placed extremely close to corridors that were served by the relay UAR, resulting in a mean DT of just 24.3 seconds in version 5 when using Method 5, compared to a mean DT across all versions of Method 5 of 60.24 seconds. Method 0 performed best in distributions 3 and 4. Method 6 has a very low mean TST for distribution 3 (Figure 6.18), but this may be misleading as the time taken to return all victims for the same distribution (Figure 6.19) is still very slow.

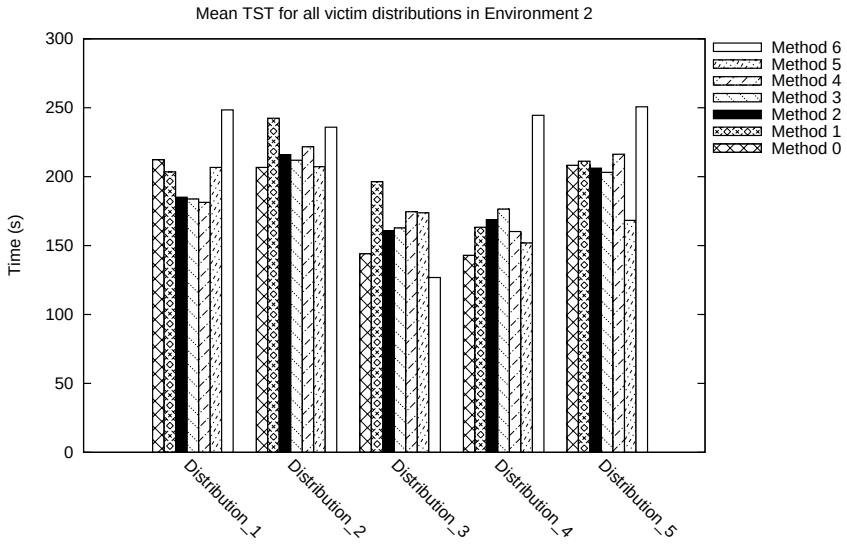


Figure 6.18: Mean Total Search Time for Environment 2

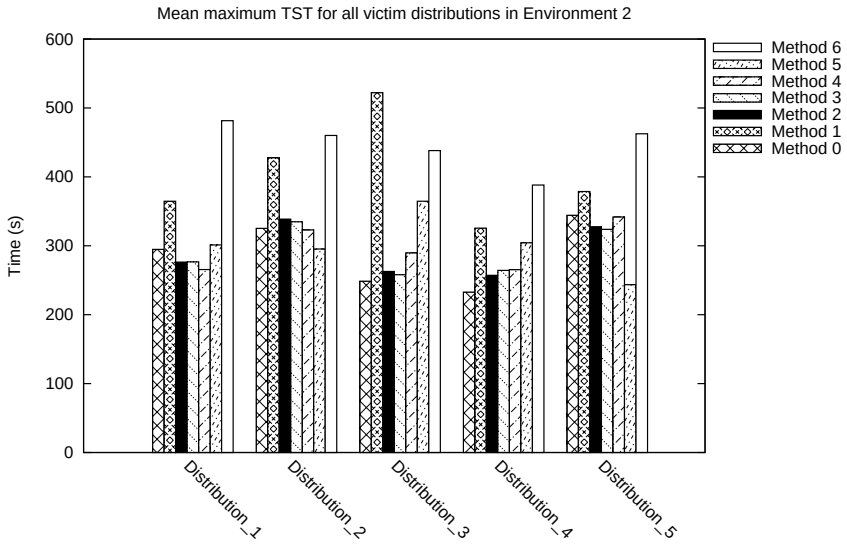


Figure 6.19: Mean maximum Total Search Time for Environment 2

	Method						
	0	1	2	3	4	5	6
First TST	116	74	113	111	103	90	114
Mean TST	183	203	186	186	191	182	221
Success Rate	100%	100%	100%	100%	100%	100%	100%
Highest rate-of-gain	2.62	2.36	2.56	2.56	2.52	2.64	2.17
Earliest 100% finish	289	311	293	292	297	302	446
Sum of sq. means (k)	277	367	293	401	306	295	430

Table 6.4: Metrics - Results for Environment 2

From results (shown in Table 6.4), we can see that Method 5 has the highest rate-of-gain, finding and returning a mean of 2.64 victims per minute; Method 0 has the earliest 100% finish, taking on average 289 seconds to find and return all victims, and also has the lowest sum of squared means, meaning that more victims are found earlier than in other methods. Comparing the methods graphically in Figure 6.17 shows the distribution of times that victims are discovered for each method; points plotted are the total time taken to find each victim and return information to the CS. The chart has been scaled to show the majority of results with greater clarity.. As already discussed, Methods 0, 2 and 3 gave very similar results (the difference between Methods 0 and 2 were virtually insignificant, with a maximum difference of 7%). Method 0 performed best on average, and given that Methods 2 and 3 were so close, it appears that the optimal solution would be to simply complete a full search of each room using Method 0.

While Method 0 performed best in two of the five distributions and also had the lowest mean search time for all victims (Figure 6.19), Methods 4 and 5 were shown to be the optimal strategy under certain conditions, performing the search task far faster than Method 0 (10% faster and 29% faster, respectively). Therefore, for this scenario the results are inconclusive. Without prior knowledge of the victim distributions, it will be impossible to predict the optimal strategy to adopt for this environment.

	Distribution				
Victim	1	2	3	4	5
1	5	5	6	5	6
2	2	0	0	5	6
3	1	6	1	5	6
4	0	6	6	5	5
5	0	5	6	0	5
6	1	0	6	1	5
7	6	0	6	0	0
8	6	6	6	0	0

Table 6.5: Victim's choice of Method for Environment 2

By looking at the data from the point of view of the victims in Table 6.5, it is possible to see that, given all of the search data, 35% of all victims would like Method 6 to have been used, despite the fact that this method performs very poorly for the majority of victims, as previously seen in Figure 6.17.

6.5.3 Environment 3

The results from Method 0 would determine the input criteria for Methods 2 and 3; the mean PTT of 56.09 seconds was found from Figure 6.20, and this was used to generate a curve in Figure 6.21, which gave an optimal PRT of 71.88 seconds. Results in Figure 6.22 show the mean TST for each victim found by each method. These figures have been normalised to take missing victims into account, but the success rates for Methods 2 and 3 were statistically close enough to 100% that this made little difference to the final graph; Method 2 found 2999 from 3000, Method 3 found 2984 from 3000 (99.47%).

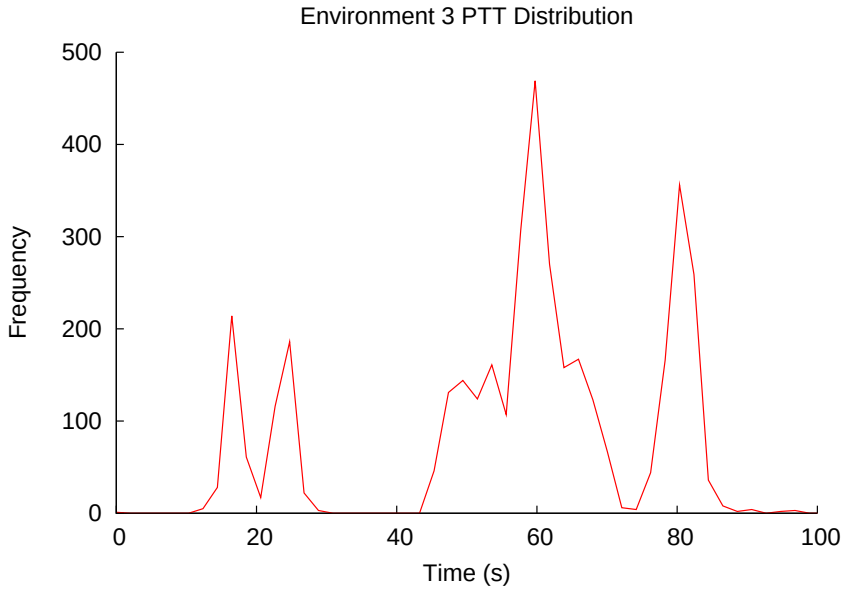


Figure 6.20: Distribution of Patch Travel Times for Environment 3

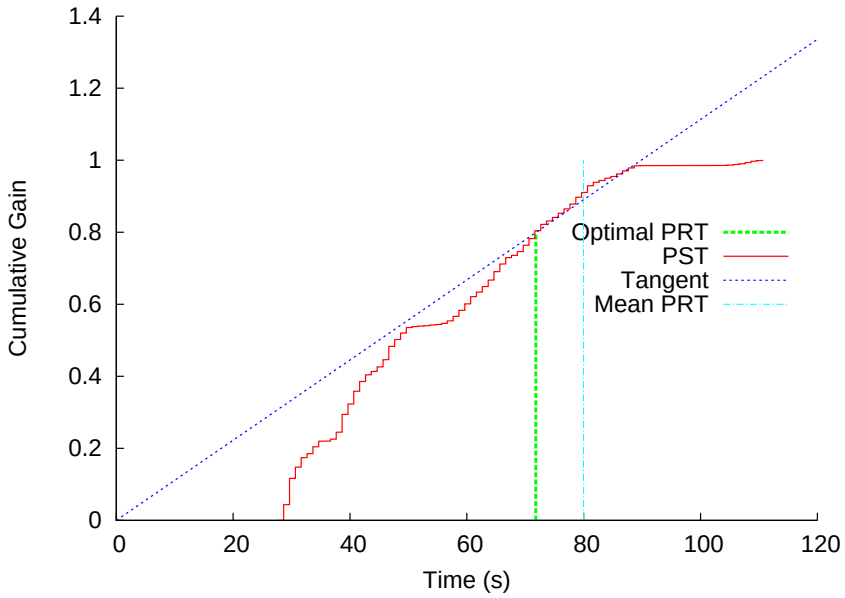


Figure 6.21: Rate-of-gain chart for PST in Environment 3

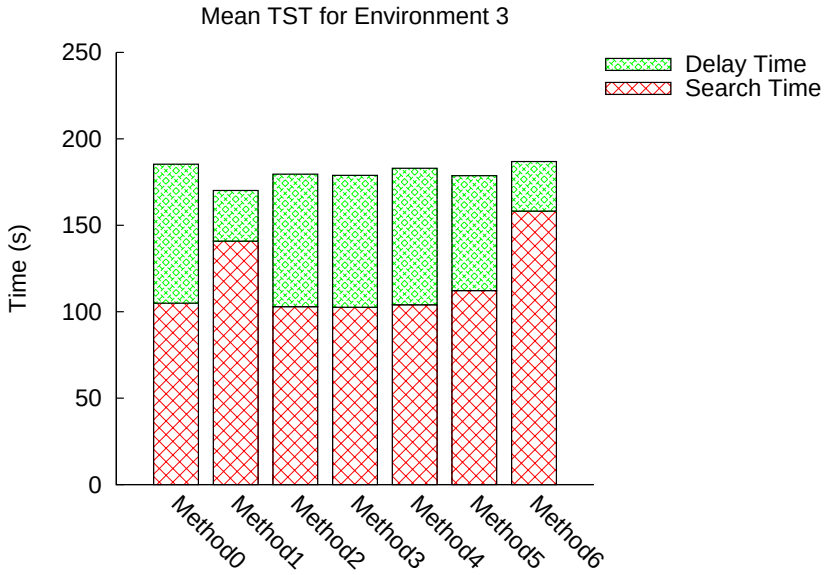
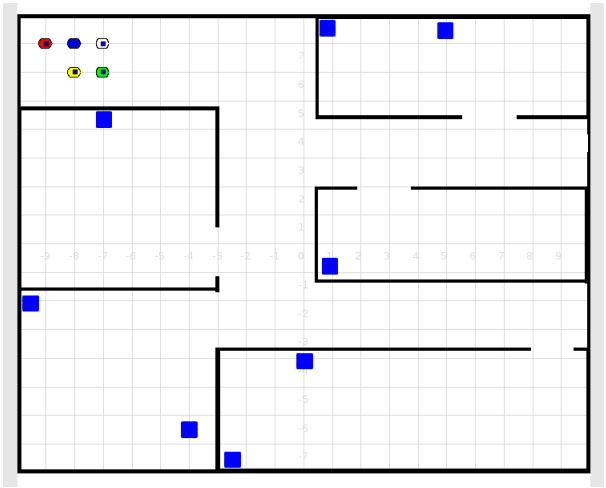
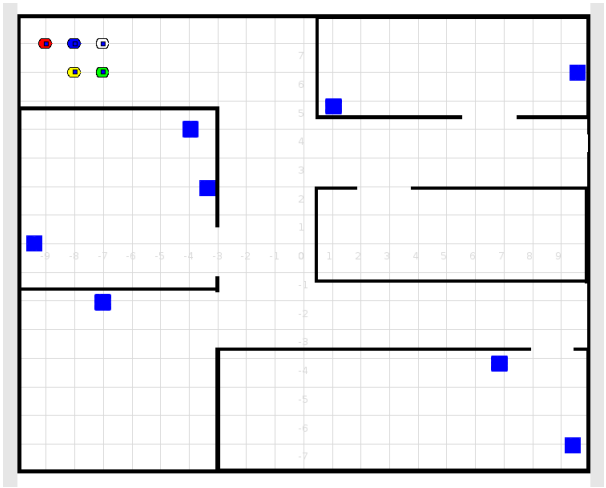


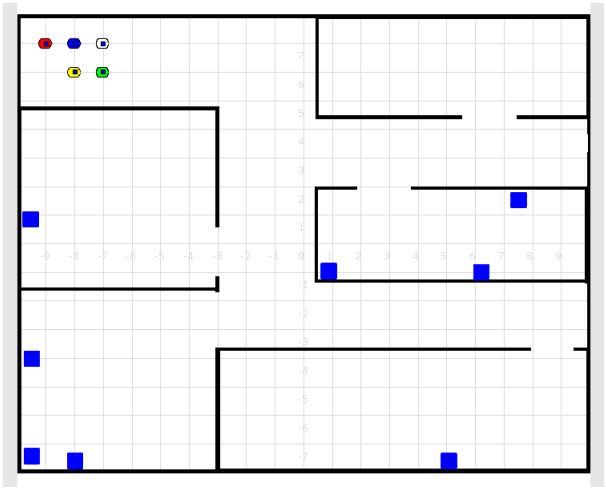
Figure 6.22: Chart showing the split between search time and delay time for each method in Environment 3



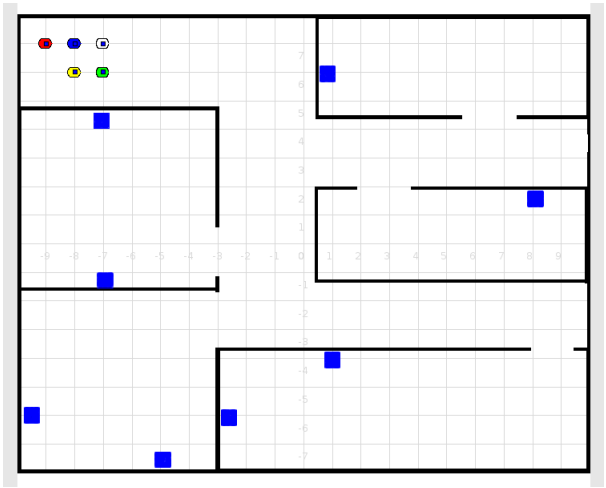
Victim Distribution 1



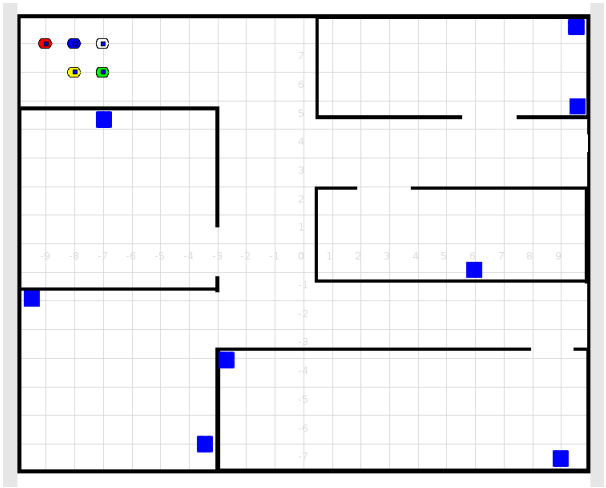
Victim Distribution 2



Victim Distribution 3



Victim Distribution 4



Victim Distribution 5

Figure 6.23: Victim distributions in Environment 3

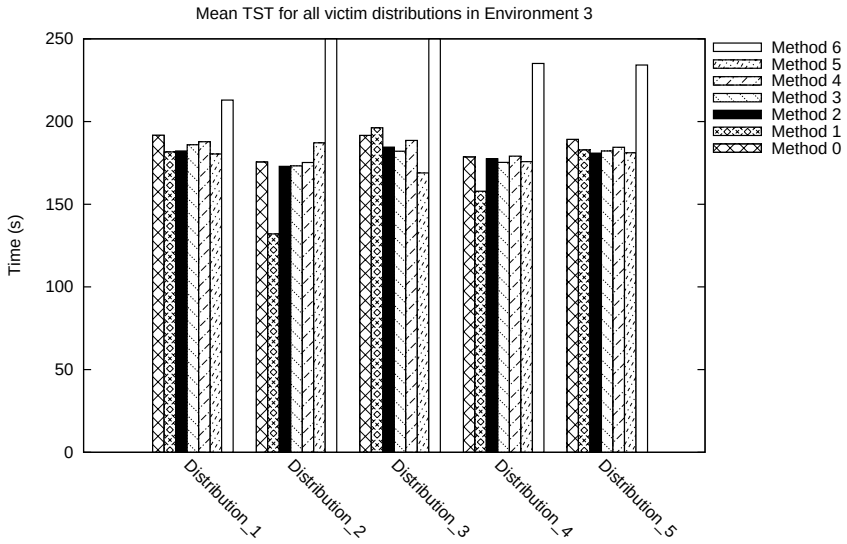


Figure 6.24: Mean Total Search Time for Environment 2

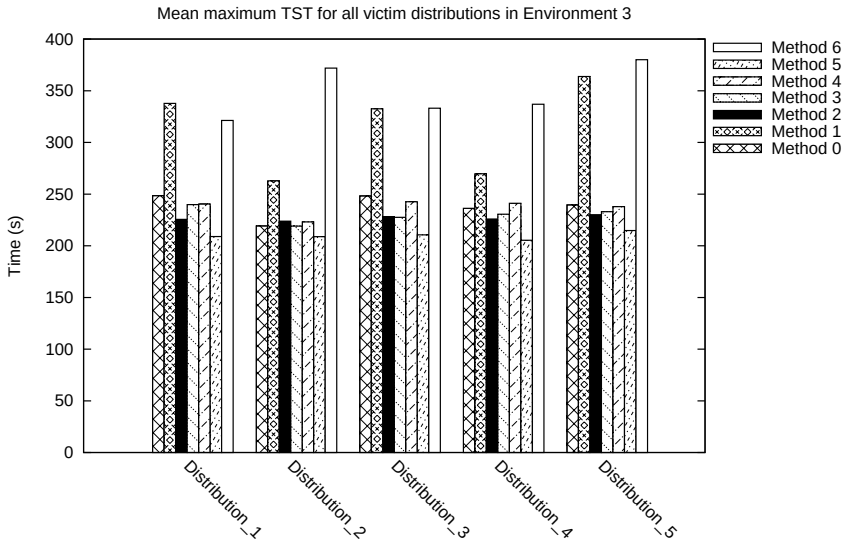


Figure 6.25: Mean maximum Total Search Time for Environment 3

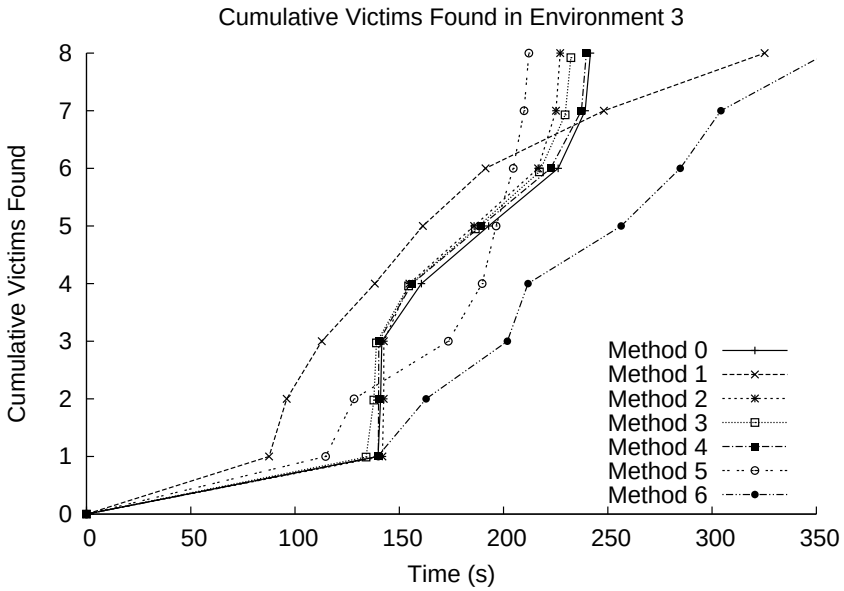


Figure 6.26: Cumulative gain curve for all methods in Environment 3

From results (shown in Table 6.6), we can see that Method 1 has the highest rate-of-gain, finding and returning a mean of 2.82 victims per minute; Method 5 has the earliest 100% finish, taking on average 210 seconds to find and return all victims, and Method 1 has the lowest sum of squared means. Method 1 also finds the first victim fastest. Figure 6.25 shows that, for the time taken to find all victims, Method 2 shows a marked improvement over Method 0 in this environment, with a time reduction of 4-10% in four of the victim distributions; in victim distribution 2 (see Figure 6.23 for a graphic of all distributions; victims have been enlarged for the sake of clarity in these diagrams), Method 0 outperformed Method 2 by 2%. Method 4 also outperformed Method 0 for distributions 1, 3 and 5, but it was Method 5 that truly stood out, with the lowest time to find all eight victims in every distribution. Figure 6.26 shows at what rate victims are discovered; points plotted are the total time taken to find each victim and return information to the CS. The chart has been scaled to show the majority of results with greater clarity. While it is true that Method 5 records a very low rate-of-gain for the first three victims (falling behind all of the other techniques except Method 6), it then still returns all victims in the fastest time. In contrast, Method 1 records the first six victims very quickly (and therefore has a low mean TST as seen in Figure 6.24), but the final two victims are reported far later than in any other method.

	Method						
	0	1	2	3	4	5	6
First TST	144	122	143	140	140	132	160
Mean TST	185	170	180	179	183	179	240
Success Rate	100%	100%	99%	100%	100%	100%	100%
Highest rate-of-gain	2.59	2.82	2.67	2.66	2.62	2.69	2
Earliest 100% finish	238	313	227	230	237	210	349
Sum of sq. means (k)	280	245	262	268	274	262	473

Table 6.6: Metrics - Results for Environment 3

Method 4's performance in Figure 6.26 is interesting; as a hybrid strategy between Methods 0 and 1, it would be natural to expect the results to lie somewhere between the two. In fact, it performs almost identically to Method 0. Method 6 performs very poorly after the first victim. Method 1 performs very well for the first six victims, but is then pipped at the post by other methods; this could be due to the nature of the environment, where all of the rooms are to one side of the CS. Method 5 stands out, returning all victims up to 19% faster than Method 0; the remaining methods all give very similar results to each other.

	Distribution				
Victim	1	2	3	4	5
1	6	6	1	3	1
2	1	1	1	1	1
3	5	1	5	1	1
4	6	1	5	5	5
5	6	1	5	1	5
6	5	1	5	5	1
7	1	0	5	6	6
8	1	1	1	1	1

Table 6.7: Victim's choice of Method for Environment 3

It is also important to look at the search from the point of view of each individual victim. If we were those victims, which method would we want to be deployed? By looking at the TST for individual victims across all Methods and victim distributions, the Method with the lowest TST was recorded in Table 6.7. The high incidence of Method 1 (52.5%) indicates that this is the strategy that most victims would wish to be deployed, yet has been shown to be inferior to other methods when looking at the overall search.

6.6 Summary

The application of the Marginal Value Theorem (MVT) from the field of ecology to the field of Urban Search and Rescue (USAR) does not prove to be particularly useful. Partly this is because the nature of USAR scenarios, which do not fit precisely onto ecological foraging strategy; there are too few victims to create a smooth, accurate curve (there are usually a large number of prey items in ecological applications of MVT). In addition, the only way to generate the curve is to perform many searches of the environment in order to get data about it; this is a major drawback. It may be possible to generate this data automatically, based on some input such as number and size of rooms and expected number of victims. In the experiments conducted so far, the application of MVT did not prove of sufficient benefit to justify its application, given the drawbacks. It did however prevent any one robot remaining in one room for too long; as humans would conduct a brief search of each room initially, so a brief search by robot could be useful prior to deploying people. Finally, and most crucially, in a real-life USAR scenario it is absolutely vital that a victim is not missed. As such, completing each room search is preferable to saving a small amount of time and potentially missing a victim. Therefore, the application of MVT in order to optimise victim search in USAR has not proven to be a useful strategy so far. However, the limited number of environments tested so far leaves room for MVT to show its merits in future work. That said, Method 2 had very high rates of finding victims in all three environments, indicating that the optimal PRT for each environment was accurate and that the overall optimal strategy was simply to search the entirety of each room. MVT therefore is applicable, but offers no tangible benefit over Method 0, while having the drawback of requiring substantial patch data in order to generate an optimal PRT.

In our experiments, our hypothesis was that opportunistic communication would be the best solution to the problem. This ran in contradiction to our proof-of-concept experiment in Section 4.2.2, which showed that a single dedicated relay UAR (Methods 1 and 4) outperformed a group of randomly moving UARs that exchange data only through opportunistic contact (Methods 0, 2, 3). The reason for this hypothesis was because, in our proof-of-concept experiment we only had a single victim to report; by adding further victims, it was expected that the time taken to report each victim would cause Methods 1 and 4 to record slower overall times than Methods 0, 2 and 3. This was true for the vast majority of cases (the exception being an anomaly in Environment 2, victim distribution 1, where Method 4 proved to be the best; see Section 6.5.2). Hence, in unknown

environments with multiple victims, UARs performing a full search and communicating opportunistically outperforms using either one or two dedicated relays. However, it became apparent that, when the environmental conditions are suitable (see Section 6.5.3), using a single dedicated stationary UAR to relay data can lead to a significantly reduced search time compared to pure search strategies, despite the loss of a search UAR.

An analysis of the metrics used to compare the methods shows that different methods tend to do better for each metric. In order to find which is the superior method, it is therefore necessary to compare metrics. While high rate-of-gain, mean TST or sum of squared means all give an indication of the overall pattern of when victims will be returned, it is earliest 100% finish that is probably the most vital metric, since it not only means that all victims are treated more equally, but also it marks the end of the search process, enabling the CS to switch their focus to recovery of victims and monitoring hazards. In our experiments, this means that Methods 0 and 2 were the best in Environment 1, Method 5 was best in Environment 3, and the results were inconclusive for Environment 2, with a choice between Methods 0, 2, 4 and 5 (Method 4 being an anomaly as already discussed). Given that MVT (Method 2) effectively returned the same as a straight-forward area division search (Method 0) in our experiments, we can simplify by saying that it comes to a choice between an area division strategy or use of a stationary relay.

Analysis of the lowest TST for each victim in each distribution found that, generally, it was those methods that performed worst overall in terms of finding and returning all victim coordinates that tended to be best for any one individual. As such, it would be natural for any of the victims to prefer that method to be deployed, despite the fact that Methods 3, 6 and 1 all performed very poorly after the first few victims; Method 3 in Environment 1, in particular, failed to even find 30% of the victims. Yet it remains the case that, for a majority of victims in a majority of distributions, these methods were the fastest to recover them. It is therefore necessary to remain objective and select the method that returns all victims in the lowest amount of time. This is an important finding; incident commanders have to make decisions under stress, and with victim's families pressuring them, it is possible for mistakes to be made. Choosing a method based on the wishes of the victims may prove detrimental to a greater number of people. It is an important consideration that the wishes of an individual victim - or even a group of victims - can contradict the view of helping all victims with maximum speed. Whether or not it is better to recover the majority of victims in a faster time at

the expense of the minority of victims who may not be found for a considerable amount of time is an area for philosophical discussion, but there is certainly room for further research that looks at how to align these two viewpoints.

6.7 Future Work

There is a need to analyse the characteristics of the environment in order to establish which technique is most appropriate. Qualitatively, it can be seen that if the CS is out of line-of-sight from the majority of rooms (as in Environment 3), then using a relay may be appropriate. Likewise, if CS has a good reach across the main corridor (such as in Environment 1) then area division seems optimal. Quantitively, a measure of this could be based on the number of rooms divided by the number of corridors; a high number is indicative of area division being appropriate (since the CS can potentially cover a larger number of rooms), while a low number is indicative of a relay being more appropriate (as seen in Environment 3). Equally, a low number of rooms is ideal for the application of a relay, where fewer UARs are available to perform the search, while a large number of rooms might favour an area division approach. However, these are fairly simplistic metrics; further work would be required to create other environments where particular characteristics of the environment are carefully selected and altered in precise ways to find the resulting difference in the performance of each method. It would be a simple process to design an environment that fits one or another technique; as such, the author recommends that future research is based on a single environment which has its features altered, such as moving the doorways and measuring the distance between doorways as a metric. One particular feature that would need analysis would be buildings with multiple entry points. Another would be to allow victims to be placed within corridors, since MVT is no longer applicable. In addition, most of our environments have been fairly similar sized; were the environment much larger then other strategies might be favoured. The same applies if more (or fewer) UARs were deployed, especially if the number of UARs is large enough to allow dedicated relays to take position without causing a delay in the search. In real-life USAR operations, search dogs are sometimes deployed. This could be modelled by having some search nodes moving around the environment, but in a way that is non-controllable. If fitted with wearable communication equipment, they could also act as relays, but cannot be retasked by the CS. The same applies to any human rescuers, who may have other tasks to deal with. Their deployment alters the ad hoc communication situation, but may not alter the search task. In short, there are numerous areas for future work by just looking at the environment and number of UARs deployed.

There is also potential for hybrid strategies. Method 1 returns many victims early in the search, resulting in very low mean TST in all three environments, and also in the cumulative gain charts (Figures 6.11, 6.17 and 6.26). It may be possible to assign a single UAR to pursue Method 1 for a period of time, then switch it to another strategy in order to utilise the quick returns of the first few victims that is characteristic of Method 1. Alternatively, it could be reassigned to act as a relay; either a stationary relay as seen in Method 5, or a roving relay similar to the concepts of Runners [16] or Message Ferries [92] as discussed in Section 2.6.2. However, it is initially unclear which of these strategies will be the best, nor is it obvious as to which of the searching UARs should act in this manner; one that starts at the furthest point from the CS, for example, or the closest? Likewise, how long should the UAR stay in Method 1 before making a switch? There is plenty of scope for further experimentation, and it is the author's opinion that any new method should be compared with Methods 0 and 5, in order to provide comparisons with this work. Likewise, teams could be deployed where each UAR follows a different strategy; analysis could then concentrate on the optimal team composition.

There are also other metrics that were not included in this scenario but might cause a change in behaviour in the searching UARs, such as the severity of the casualty; if a victim was discovered whose life signs were very weak, then this victim might be treated as being of higher importance than continuation of the search. This may in turn cause the UAR to choose to immediately return to the CS in order that the victim might be recovered faster. Likewise, the total number of victims discovered may be relevant; if one UAR discovers a large number of victims in one location, then this may also a trigger a change in behaviour. This is especially true if the environment is hazardous to the UAR and there is a risk that the UAR will become trapped or disabled, and therefore not be able to report the coordinates of victims that have already been discovered, with potentially fatal results. The distance that needs to be travelled to communicate with the CS is also a factor. Autonomous decision making would be required to allow UARs to perform correctly, but in order to capture the correct behaviour, further discussions with technical rescue teams would be required.

Chapter 7

Conclusions

Urban Search and Rescue (USAR) environments are dangerous and difficult places in which to operate, and deploying mobile autonomous robots can avoid putting rescue workers at risk. Wired communication can lead to entanglement, and wireless communication is generally restricted to line of sight, so robots will often end up outside communication range of the human-operated Command Station (CS). From the point of view of the CS, it is desirable to get victims' locations as quickly as possible. This problem is two-fold: victims must be found, and also have their location communicated to the CS. Minimising the time taken to find the victims (Search Time - ST) will result in the first victim's coordinate being delayed from reaching the CS while the robot continues to search. Minimising the time taken to send data to the CS after finding each victim (Delay Time - DT) means halting the search to communicate findings, resulting in later ST for remaining victims. The aim is to minimise the combination of ST + DT = Total Search Time (TST). Therefore, there is a trade-off between the strategy used to search the environment, and the communication strategy used to relay data.

If there are insufficient nodes (which can either be robots or wearable computers worn by humans or dogs) to cover the entire environment, then in order to explore the entire environment, nodes have no choice but to break communications links between themselves and the CS. It may not possible to calculate where and when nodes might come into contact again. There are two solutions that can be implemented: either alter the movement of the nodes to force them to come within communication range, or allow them to move wherever they want and rely on opportunistic contacts to communicate. Extending the range of the CS via the use of dedicated relays was implemented, and compared to an area division search method. Results in an open area with obstacles using a 4m communication range showed that Area Division had the fastest search time by a considerable margin, but that both techniques had large overheads when compared to a Random Walk

model; this was due to a larger number of interactions.

While it is not possible to control node interaction without affecting the speed of the search, it is possible to streamline the data exchange process. A novel algorithm for data exchange in opportunistic networks, DEM, was developed in order to reduce the number of data that are exchanged, and the amount of overhead required, when teams of cooperating agents are communicating opportunistically. DEM has lower overheads than existing methods while maintaining the propagation speed of flooding.

With DEM and an area division strategy in place, more complex indoor environments were tested using line-of-sight communication. A novel implementation of an ecological foraging strategy called the Marginal Value Theorem (MVT) was tested with a view to optimising the search process. We implemented several search strategies, along with strategies that deployed one or two nodes as dedicated relays while the others followed the area division search. Results showed that there was no straightforward way to choose the best strategy. Typically, it was either the area division search, or the use of a single dedicated relay in conjunction with an area division search, that led to the best solution. This indicates that it is these two strategies that should be explored in more depth.

MVT did offer a slight reduction in TST, but this was modest and there is the complication of needing to collect data to build the gain-curve in order to generate the optimal Patch Residence Time (PRT). Effectively, MVT predicted that the optimal strategy in the environments tested was to have a large enough PRT to finish the search entirely. MVT was developed through analysis of animals foraging, such as birds collecting berries from bushes. As such, modelling victims as food items in MVT does not fit well enough; there are too few victims, and each is too important to miss. In these sparsely occupied environments, the application of MVT, although novel, did not lead to a noticable improvement over an area division strategy.

Our experiments in Chapter 6 indicate various areas for future work. Environmental characteristics, such as number of rooms and corridors, room size, overall environment size and the combination of such features, affect which of the methods should be employed. By analysing the effects of changes to an environment, a clearer pattern should emerge of which characteristics are most relevant to the choice of search technique. Victim distribution also affected the optimal method. However, this is beyond the control of the rescue workers and therefore must

remain an uncontrollable variable. Alternative hybrid strategies can also be deployed; either completely new methods which try to use the benefits of each of the already implemented methods, or where a team is composed of nodes that follow different strategies. One example of a hybrid strategy would be switching from immediate return early in the search process and while close to the CS, to performing an area division search later on, and when further from the CS. Knowing at what point this switch should occur, and which role to switch to, is also a research topic.

Alterations to the model, such as having different categories of victim, plus hazards and fuel or power considerations could be used to develop a system capable of assessing casualties and hazards in the field, then choosing an appropriate course of action via autonomous decision making. This will allow agents deployed in USAR to assess whether it is worth abandoning their current search in order to report mission data, or to continue searching. Essentially, this will mean a step away from scripted strategies such as those discussed in this thesis, allowing the agent to switch behaviours according to current mission data, making the system more effective. Potential experiments include testing DEM in lossy environments and altering the number of robots deployed. The deployment of mobile relays, either dedicated relays or pursuing a hybridised search and communication function, is also an area for experimentation, particularly in the area where an individual robot switches function between performing the role of search and the role of relay. Finally, each component of the model can be made more realistic; variable terrain which affects movement speed, victims requiring a certain amount of time to perform recognition, and patches of the environment where communication distances or loss rates are highly variable, would all lead to a better understanding of this complex research area.

The contribution of this thesis is to show that it is not possible to divorce the search strategy from the communication strategy in USAR scenarios; the two are co-dependent. When communication is limited to line-of-sight or short range node-to-node contacts, the environmental characteristics directly effect which strategy is optimal. DEM can be used in conjunction with opportunistic inter-node relay, which results in the entire system acting as a broken ad hoc network. For large environments where there are a lot of rooms, or for very small environments, then an area division search is the best strategy. Only where the communication range of the CS is small, and where adjoining corridors could be used to significantly extend this range, then a dedicated relay can be deployed which will outperform a system where all nodes are tasked to search.

References

[1] B. Andrews, K. Passino, and T. Waite. Foraging Theory for Autonomous Vehicle Decision-making System Design. *J. Intell. Robotics System*, 49(1):39–65, 2007.

[2] J. Armstrong. Key management for network-enabled platforms in the 'system of systems' environment. *BAE Systems Proprietary*, 2006.

[3] J. Armstrong. Security threat models for semi-autonomous integrated modular systems in a network enabled environment. *BAE Systems Proprietary*, 2006.

[4] N. Atay and B. Bayazit. Emergent task allocation for mobile robots. *in Proceedings of Robotics: Science and Systems*, 2007.

[5] N. Bailey. *The mathematical theory of infectious diseases and its applications.* New York : Hafner Press, 1975.

[6] M. Batalin and G. Sukhatme. Coverage, exploration and deployment by a mobile robot and communication network. In *Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks*, pages 376–391, 2003.

[7] C. Becker and G. Schiele. New mechanisms for routing in ad hoc networks through world models. In *4th Plenary Cabernet Wksp. '05*, 2001.

[8] M. Begon. *Ecology: from individuals to ecosystems.* Wiley-Blackwell, 4, illustrated, revised edition, 2006.

[9] B. Bellur, M. Lewis, and F. Templin. An ad-hoc network for teams of autonomous vehicles. *Proc. First Annual Symposium on Autonomous Intelligence*, 2002.

[10] J. Blitch. Artificial intelligence technologies for robot assisted urban search and rescue. *Expert Systems with Applications*, 11(2):109 – 124, 1996. Army Applications of Artificial Intelligence.

[11] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187, - 1989.

[12] D. J. Bruemmer. Dynamic autonomy for urban search and rescue. *AAAI Robotics Competition and Exhibition*, 2002.

[13] W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 21(3):376–386, June 2005.

[14] J. Casper, M. Micire, and R. Murphy. Issues in intelligent robots for search and rescue. volume 4024, pages 292–302. SPIE, 2000.

[15] E. Charnov. Optimal foraging, the marginal value theorem. *Theoretical Population Biology*, 9(2):129–136, April 1976.

[16] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. An efficient communication strategy for ad-hoc mobile networks. In *PODC '01: Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 320–322, New York, NY, USA, 2001. ACM.

[17] J. Chen, S. Wolfe, and S. Wragg. A distributed multi-agent system for collaborative information management and sharing. In *CIKM '00: Proceedings of the ninth international conference on Information and knowledge management*, pages 382–388, New York, NY, USA, 2000. ACM.

[18] X. Chen, H. Zhai, J. Wang, and Y. Fang. A survey on improving tcp performance over wireless networks. In *In Resource Management in Wireless Networking, Cardei M, Cardei I, Du D-Z*, pages 657–695. Kluwer Academic Publishers, 2005.

[19] Z. Chen, H. Kung, and D. Vlah. Ad hoc relay wireless networks over moving vehicles on highways. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 247–250, New York, NY, USA, 2001. ACM.

[20] N. Christoffersson and C. Backstrom. Urban search and rescue - an evaluation of technical search equipment and methods. page 55, 2006.

[21] C. Clark. Dynamic robot networks: A coordination platform for multi-robot systems. *PhD Thesis, Stanford.*

[22] S. Cooper. Southern Fire Station Leicestershire Fire Service, 2009. Interview.

[23] V. Crespi, W. Chung, and A. Jordan. Decentralized sensing and tracking for uav scheduling. volume 5403, pages 451–462. SPIE, 2004.

[24] UK Statue Law Database. The fire and rescue services (emergencies) (england) order 2007 (no. 735), 2007.

[25] F. Dechaume-Moncharmont, A. Dornhaus, A. Houston, J. McNamara, E. Collins, and N. Franks. The hidden cost of information in collective foraging. *Proceedings of the Royal Society B: Biological Sciences*, 272(1573):1689–1695, 2005.

[26] A. Demers, K. Petersen, M. Spreitzer, D. Terry, M. Theimer, and B. Welch. The bayou architecture: Support for data sharing among mobile users. In *Proceedings IEEE Workshop on Mobile Computing Systems & Applications*, pages 2–7, Santa Cruz, California, 8-9 1994.

[27] S. O Donnell. Southern Fire Station Leicestershire Fire Service, 2009. Interview.

[28] A. Doria. Providing connectivity to the saami nomadic community. In *In Proc. 2nd Int. Conf. on Open Collaborative Design for Sustainable Innovation*, 2002.

[29] Draeger. Merlin. *http://www.draeger.com*, 2010.

[30] H. Durrant-Whyte. Beginner's guide to decentralised data fusion. *Australian Centre for Field Robotics, The University of Sydney NSW*, July 2000.

[31] M. Frisby. Southern Fire Station Leicestershire Fire Service, 2009. Interview.

[32] A. Gage, R. Murphy, E. Rasmussen, and B. Minten. Shadowbowl 2003 [simulated mass-casualty exercise]. *Robotics & Automation Magazine, IEEE*, 11(3):62–69, Sept. 2004.

[33] B. Gerkey, R. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *In Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, 2003.

[34] K. Harras, K. Almeroth, and E. Belding-royer. Delay tolerant mobile networks (dtmns): Controlled flooding schemes in sparse mobile networks. In *In IFIP Networking*, 2005.

[35] A. Hayes and D. Wilson. Peer-to-peer information sharing in a mobile ad hoc environment. *Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on*, pages 154–162, Dec. 2004.

[36] Hokuyo. Urg-04lx. *http://www.hokuyo-aut.jp*, 2009.

[37] A. Houston, J. McNamara, and J. Hutchinson. General results concerning the trade-off between gaining energy and avoiding predation. *Royal Society of London Philosophical Transactions Series B*, 341:375–397, sep 1993.

[38] A. Howard, M. Matadd, and G. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots*, 13:113–126, 2002. 10.1023/A:1019625207705.

[39] M. Hsieh, A. Cowley, V. Kumar, and C.J. Taylor. Towards the deployment of a mobile robot network with end-to-end performance guarantees. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2085–2090, May 15-19, 2006.

[40] Innowebtive. Koala. *http://www.innowebtive.com/kteam/robots/koala/index.html*, 2002.

[41] A. Jacoff, E. Messina, B. Weiss, S. Tadokoro, and Y. Nakagawa. Test arenas and performance metrics for urban search and rescue robots. *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 4:3396–3403 vol.3, Oct. 2003.

[42] JAW. Search and rescue missions. *http://www.justaeroworks.com/pages/JAW.08SRM.html*, 2008.

[43] J. Jennings, G. Whelan, and W. Evans. Cooperative search and rescue with a team of mobile robots. *Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference on*, pages 193–200, 7-9 Jul 1997.

[44] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. pages 96–107, 2002.

[45] A. Kleiner, E. Kleiner, J. Prediger, and B. Nebel. RFID technology-based exploration and slam for search and rescue. pages 4054–4059, 2006.

[46] V. Kumar, G. Bekey, and A. Sanderson. Networked robots. *WTEC Panel on INTERNATIONAL ASSESSMENT OF RESEARCH AND DEVELOPMENT IN ROBOTICS DRAFT REPORT*, pages 57–72, August 2005.

[47] M. Lewis, K. Sycara, and I. Nourbakhsh. Developing a testbed for studying human-robot interaction in urban search and rescue. In *Proceedings of the 10th International Conference on Human Computer Interaction*, pages 22–27, 2003.

[48] M. Li, W. Chien, and A. Sivasubramaniam. Efficient peer-to-peer information sharing over mobile ad hoc networks. In *In MobEA*, 2004.

[49] A. Makarenko, A. Brooks, S. Williams, H. Durrant-Whyte, and B. Grocholsky. A decentralized architecture for active sensor networks. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 2:1097–1102 Vol.2, April 26-May 1, 2004.

[50] A. Makarenko and H. Durrant-Whyte. Decentralized data fusion and control in active sensor networks. In *In Proceedings of the Seventh International Conference on Information Fusion*, 2004.

[51] L. Matthies, Y. Xiong, R. Hogg, D. Zhu, A. Rankin, B. Kennedy, M. Hebert, R. Maclachlan, C. Won, T. Frost, G. Sukhatme, M. McHenry, and S. Goldberg. A portable, autonomous, urban reconnaissance robot. *Robotics and Autonomous Systems*, 40(2-3):163 – 172, 2002.

[52] E. Messina and A. Jacoff. Performance standards for urban search and rescue robots. volume 6230, page 62301V. SPIE, 2006.

[53] E. Messina, A. Jacoff, J. Scholtz, C. Schlenoff, H. Huang, A. Lytle, and J. Blitch. Statement of requirements for urban search and rescue robot performance standards preliminary version. pages 19–20, May 2005.

[54] M. Micire. Analysis of the robotic assisted search and rescue response to the world trade centre disaster. *Masters Thesis, University of South Florida*, 2002.

[55] J. Mitchell and A. Sparks. Communication issues in the cooperative control of unmanned aerial vehicles. *First Annual Allerton Conference on Communication, Control*, 2003.

[56] K. Murphy, R. Schultz, and A. Osuka. Usar competitions for physically situated robots. In *Robotics & Automation Magazine, IEEE*, volume 9, pages 26–33, 2002.

[57] R. Murphy, J. Casper, J. Hyams, M. Micire, and B. Minten. Mobility and sensing demands in usar. *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Conference of the IEEE*, 1:138–142, 2000.

[58] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *WOWMOM '05: Proceedings of the*

Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks, pages 183–189, Washington, DC, USA, 2005. IEEE Computer Society.

[59] D. Nain, N. Petigara, and H. Balakrishnan. Integrated routing and storage for messaging applications in mobile ad hoc networks. In *In Proc. of Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt03), Sophia Antipolis*, pages 595–604, 2003.

[60] W. Ng, B. Ooi, K. Tan, and A. Zhou. Peerdb: A p2p-based system for distributed data sharing. *19th International Conference on Data Engineering*, pages 633–644, March 2003.

[61] H. Nguyen, H. Everett, N. Manouk, and A. Verma. Autonomous mobile communication relays. volume 4715, pages 50–57. SPIE, 2002.

[62] H. Nguyen, N. Farrington, and N. Pezeshkian. Maintaining communication link for tactical ground robots. In *AUVSI Unmanned Systems North America 2004*, pages 3–5. America, 2004.

[63] H. Nguyen, N. Pezeshkian, A. Gupta, and N. Farrington. Maintaining communication link for a robot operating in a hazardous environment. In *ANS 10 th Int. Conf. on Robotics and Remote Systems for Hazardous Environments*, pages 28–31, 2004.

[64] H. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, and J. Spector. Autonomous communication relays for tactical robots. In *in Proceedings of the International Conference on Advanced Robotics (ICAR)*, pages 35–40, 2003.

[65] L. Parker. Generating self-reliant teams of autonomous cooperating robots: Desired design characteristics. *Autonomy Control Software Workshop, Agents '99, Seattle, WA, May 1,1999*, 1999.

[66] T. Pavlic and K. Passino. Foraging theory for autonomous vehicle speed choice. *Eng. Appl. Artif. Intell.*, 22(3):482–489, 2009.

[67] B. Pimentel and M. Campos. Cooperative communication in ad hoc networked mobile robots. *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 3:2876–2881 vol.3, 27-31 Oct. 2003.

[68] RAF. Search and rescue - our role at home. *http://www.raf.mod.uk/careers/abouttheraf/searchandrescue.cfm*, 2008.

[69] A. Ray. *Cooperative Robotics using Wireless Communication*. 2005.

[70] J. Redi and J. Bers. Exploiting the interactions between robotic autonomy and networks. *Second International Workshop on Multi-Robot Systems*, 2003.

[71] M. Rehmani, A. Viana, H. Khalife, and S. Fdida. A Cognitive Radio Based Internet Access Framework for Disaster Response Network Deployment. Research Report RR-7285, INRIA, 05 2010.

[72] K. Remley, G. Koepke, E. Messina, and A. Jacoff. Standards development for wireless communications for urban search and rescue robots. pages 66–70, Feb 2007.

[73] M. Richards, D. Whitley, J. Beveridge, T. Mytkowicz, D. Nguyen, and D. Rome. Evolving cooperative strategies for uav teams. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1721–1728, New York, NY, USA, 2005. ACM.

[74] M. Rooker and A. Birk. Combining exploration and ad-hoc networking in robocup rescue. In Daniele Nardi, Martin Riedmiller, and Claude Sammut, editors, *RoboCup 2004: Robot Soccer World Cup VIII*, volume 3276 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages pp.236–246. Springer, 2005.

[75] S. Roumeliotis and G. Bekey. Distributed multirobot localization. *Robotics and Automation, IEEE Transactions on*, 18(5):781–795, Oct 2002.

[76] S. Roumeliotis and M. Mataric. "small-world" networks of mobile robots. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, page 1093. AAAI Press / The MIT Press, 2000.

[77] N. Sanderson. Knowledge management in mobile ad-hoc networks for rescue scenarios. *Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications, ISWC*, 2004.

[78] D. Scheidt and J. Stipes. Cooperating unmanned vehicles. *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*, pages 326–331, 19-22 March 2005.

[79] S. Sen, M. Sekaran, and J. Hale. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 426–431, Seattle, WA, 1994.

[80] SourceForge. Usarsim. *http://usarsim.sourceforge.net*, 2010.

[81] A. Speranzon. On control under communication constraints in autonomous multi-robot systems. *PhD Thesis, KTH Stockholm, Sweden*.

[82] A. Speranzon and K. Johansson. On some communication schemes for distributed pursuit-evasion games. *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, 1:1023–1028 Vol.1, 9-12 Dec. 2003.

[83] J. Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara. User mobility for opportunistic ad-hoc networking. In *WMCSA '04: Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pages 41–50, Washington, DC, USA, 2004. IEEE Computer Society.

[84] F. Tchakountio and R. Ramanathan. Tracking highly mobile endpoints. In *Proc. ACM Workshop on Wireless Mobile Multimedia (WoWMoM), Jul 2001*, 2001.

[85] J. Tower and T. Little. A proposed scheme for epidemic routing with active curing for opportunistic networks. In *AINAW '08: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops*, pages 1696–1701, Washington, DC, USA, 2008. IEEE Computer Society.

[86] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, 2000.

[87] Wang, Zhelong, Gu, and Hong. A review of locomotion mechanisms of urban search and rescue robot. *Industrial Robot: An International Journal*, 34(5):400–411, 2007.

[88] S. Whiteley, D. Waters, G. Hayward, S. Pierce, and I. Farr. Wireless recording of the calls of rousettus aegyptiacus and their reproduction using electrostatic transducers. *Bioinspiration & Biomimetics*, 5(2):026001, 2010.

[89] A. Winfield. Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. *Distributed Autonomous Robotic Systems*, 4:273–282, 2000.

[90] C. Yu, Y. Chen, L. Chen, C. Yu, Y. Chen, and L. Chen. Effective file transfer for opportunistic networks. *IEEE Infocom07*, 2006.

[91] G. Zhang and Q. Jin. Scalable information sharing utilizing decentralized p2p networking integrated with centralized personal and group media tools. *aina*, 2:707–711, 2006.

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

Appendix A

Additional Data

A.1 Total Search Time Distributions

Due to many repeated runs for each experiment in Chapter 6, it is necessary to show the variance in terms of frequency distributions. These are shown graphically in the following sections.

A.1.1 Environment 1

Results in Figure A.1 for Methods 0, 2 and 4 show that most victims' TSTs are grouped in spikes around 120 seconds and 170 seconds. Method 1 differs in being more spread out; this is due to its search pattern which gives preference to those victims found early at the expense of those not yet found, who will then end up with a high TST. Methods 5 and 6 used one or more data relays; this means that there were fewer UARs actively searching, which would explain the shift of the spikes seen for other methods.

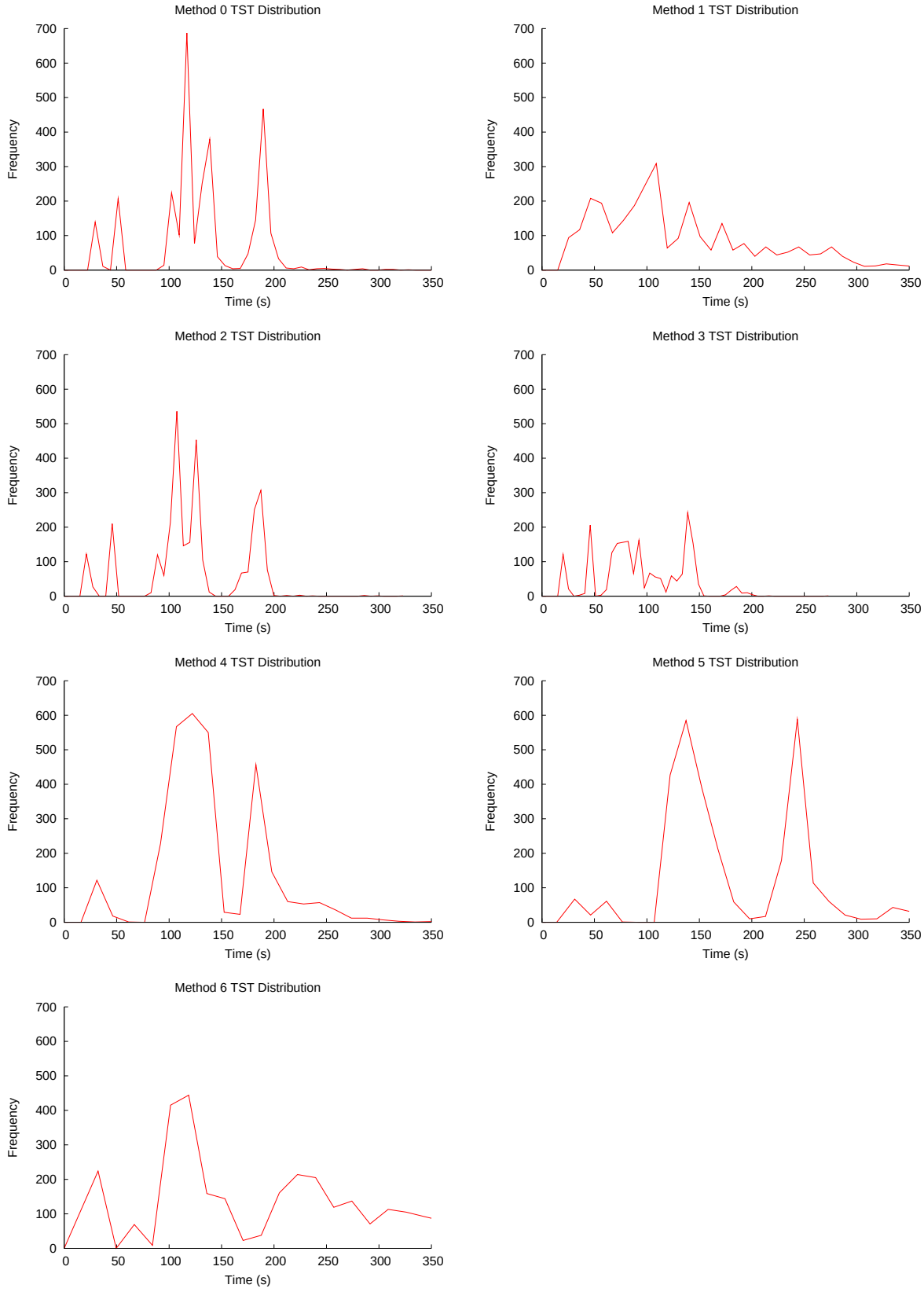


Figure A.1: TST Distributions for all methods in Environment 1

A.1.2 Environment 2

Figure A.2 shows the frequency distribution of each method; all methods have initial returns after approximately 25 seconds, then a spike around 100 seconds, where a large number of victims are found, then results tail off until 200-300 seconds, when the majority of victims are found. Both Methods 1 and 5 have a high frequency of victims reported before 100 seconds, which is a useful property as it means that the CS can begin to plan rescues for those victims reported early while the search continues. Methods 5 and 6 are far more evenly spread than the other methods.

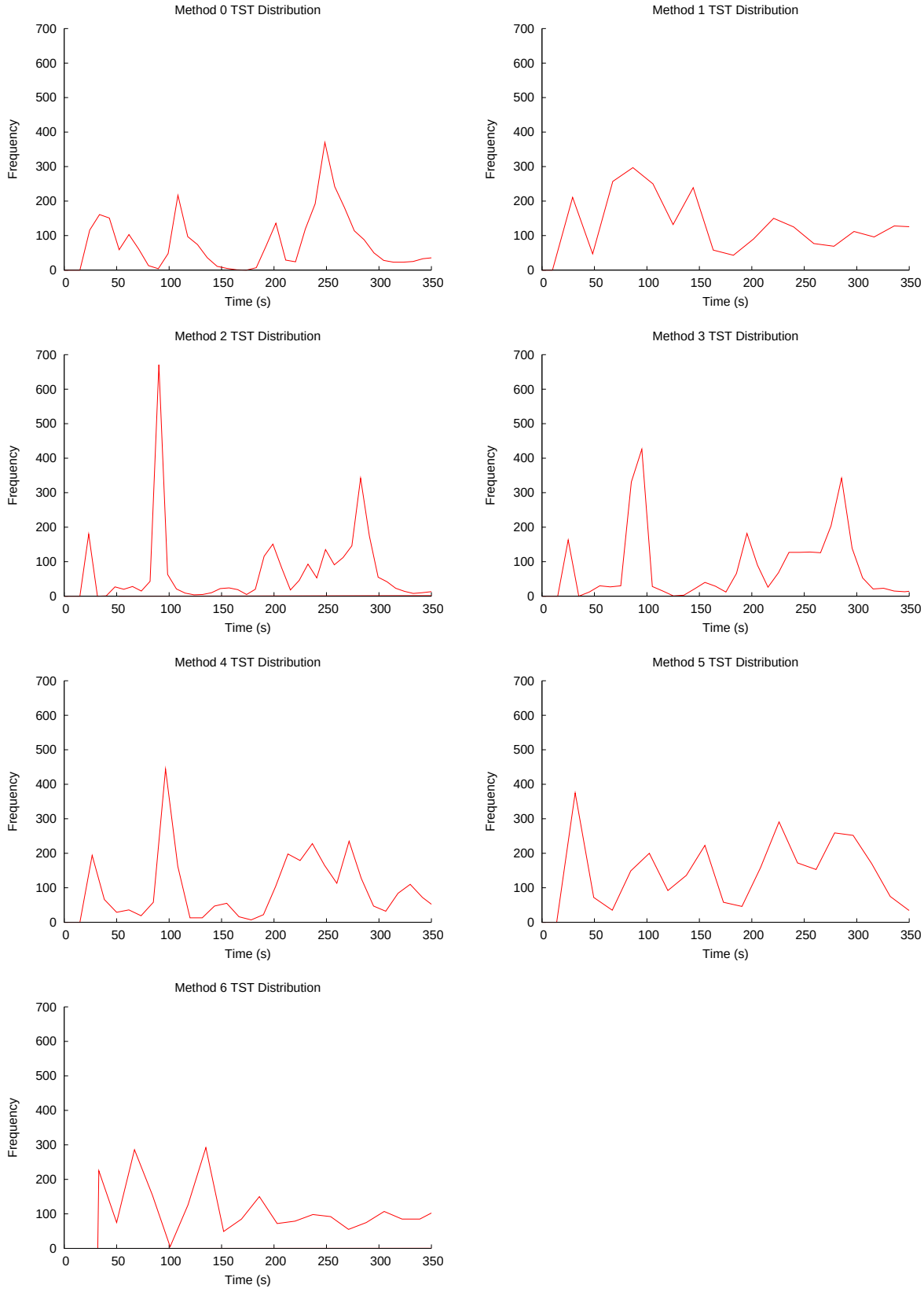


Figure A.2: TST Distributions for all methods in Environment 2

A.1.3 Environment 3

The frequency distributions in Figure A.3 show two large spikes at approximately 150 and 225 seconds for Methods 0,2,3 and 4; this reflects the fact that, in this environment, the CS is out of range of all rooms. Method 1 returns a large number of victims slightly earlier, at around 100 seconds (as also seen in Figure 6.26), but then the TST values are fairly evenly distributed for Method 1; in contrast, all other techniques have completed their searches by the 250 second mark. Method 5, having only three UARs available for the search, also shows similar spikes to Methods 0, 2, 3 and 4, except slightly earlier, at around 100 and 200 seconds; the use of a dedicated relay in this environment giving a clear reduction in TST values, despite the loss of a search UAR.

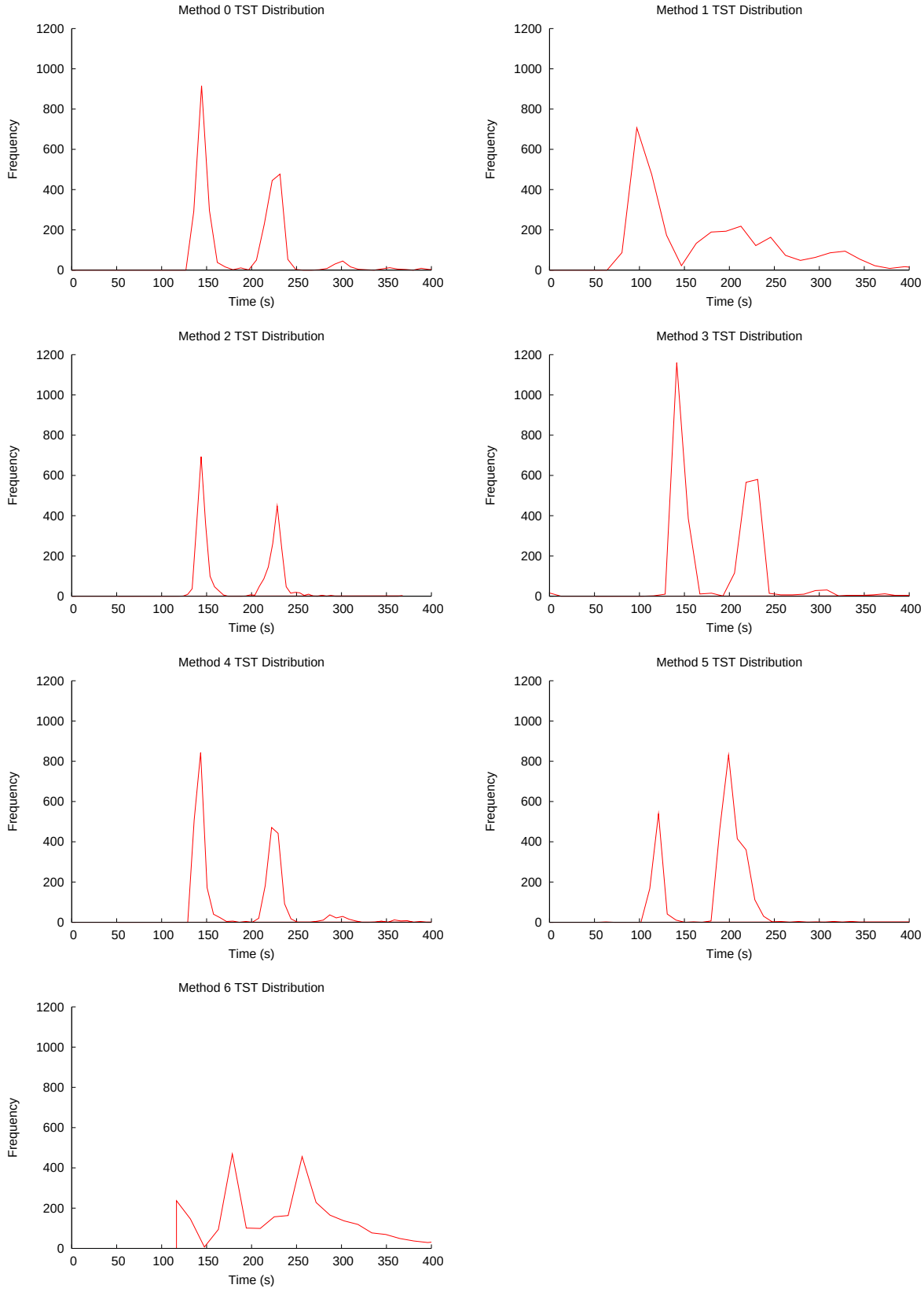


Figure A.3: TST Distributions for all methods in Environment 3

	Method						
Victim	0	1	2	3	4	5	6
1	132.85	99.39	125.64	123.25	137.71	190.23	103.19
2	19.95	26.55	19.71	21.15	31.84	292.77	215.63
3	107.57	90.31	106.43	107.87	125.24	131.68	226.36
4	186.08	229.21	182.25	138.12	180.28	178.81	296.01
5	186.08	267.72	182.25	176.74	198.48	235.64	318.76
6	113.09	123.88	105.41	106.79	112.87	129.35	100.29
7	45.88	54.36	45.93	44.44	99.41	125.01	100.97
8	132.83	49.15	124.59	106.89	128.49	190.23	24.4

Table A.1: Mean TST for all methods in Environment 1, Distribution 1

	Method						
Victim	0	1	2	3	4	5	6
1	128.64	129.59	113.44	68.77	150.56	178.95	65.59
2	128.64	39.73	113.43	68.77	149.64	178.95	24.2
3	180.59	195.37	181.35	138.6	206.65	236.88	303.65
4	179.68	167.88	175.35	136.44	178.68	238.21	362.85
5	106.85	60.56	105.96	88.91	107.81	49.89	149.27
6	106.85	120.61	105.96	89.07	107.81	124.44	207.01
7	106.85	103.13	105.96	89.07	107.81	124.4	203.24
8	107.09	94.35	105.8	N/A	106.43	128.83	97.87

Table A.2: Mean TST for all methods in Environment 1, Distribution 2

A.2 Victim Choice

A.2.1 Environment 1

The data in Table 6.3 in Section 6.5.1 is comprised of the Method with the lowest mean TST for each victim and distribution. Full data tables are shown here, with all mean TST values for all Methods shown for each distribution. 'N/A' is shown where a particular method did not find a specific victim during any of the experiments, and therefore that victim does not have a TST value.

	Method						
Victim	0	1	2	3	4	5	6
1	94.39	144.36	93.61	72.44	91.79	258.13	234.67
2	94.39	96.13	93.61	N/A	91.79	258.13	235.77
3	94.39	163.24	93.61	72.44	91.79	258.13	234.19
4	17.67	26.51	19.05	17.25	52.76	33.16	16.72
5	130.68	101.91	121.85	N/A	134.77	154.96	103.95
6	182.91	204.28	181.49	128.01	180.45	167.63	272.51
7	174.76	293.05	179.85	137.17	206.32	242.51	368.37
8	182.91	247.95	181.51	132.04	180.45	175.03	274.12

Table A.3: Mean TST for all methods in Environment 1, Distribution 3

	Method						
Victim	0	1	2	3	4	5	6
1	126.25	70.17	121.77	70.75	150.05	176.81	102.49
2	126.25	119.73	121.79	N/A	150.05	176.81	102.68
3	98.8	48.2	92.44	64.84	92.13	185.15	144.52
4	180.41	243.4	183.23	145	219.8	232.25	337.84
5	179.96	349.56	176.7	137.66	176.96	240.61	377.61
6	47.27	54.59	45.91	43.99	111.37	124.52	106.07
7	47.33	87.08	45.92	43.99	112.31	126.17	106.36
8	107.2	129.43	108.6	79.93	113.17	119.24	109.04

Table A.4: Mean TST for all methods in Environment 1, Distribution 4

	Method						
Victim	0	1	2	3	4	5	6
1	126.6	41.72	122.47	69.25	138.43	183.47	101.35
2	126.61	132.45	122.47	N/A	138.44	183.48	104.65
3	111.09	69.2	91.49	62.96	93.29	281.65	238.84
4	176.87	176.88	162.65	120.6	139.52	237.95	340.59
5	181.07	243.85	185.88	130.04	200.27	183.07	260.99
6	108.25	119.45	110.33	81.65	108.09	120.76	149.88
7	108.25	120.72	110.33	N/A	108.09	131.2	136.87
8	108.07	104.01	106.63	N/A	107.21	126.56	210.95

Table A.5: Mean TST for all methods in Environment 1, Distribution 5

	Method						
Victim	0	1	2	3	4	5	6
1	261.87	348.88	229.27	224.67	246.49	164.53	231.16
2	247.4	124.24	54.55	56.24	54.57	301.28	481.6
3	192.67	67.31	217.89	212.83	207.05	162.89	222.57
4	244.61	364.56	276.31	276.6	265.56	268.31	367.65
5	243.75	292.45	276.29	275.05	259.28	268.17	367.64
6	106.4	61.49	85.13	85.27	85.79	95.37	68.48
7	106.4	125.71	85.13	85.28	85.79	95.51	69.12
8	294.85	243.25	256.23	254.81	246.41	297.4	179.39

Table A.6: Mean TST for all methods in Environment 2, Distribution 1

	Method						
Victim	0	1	2	3	4	5	6
1	217.21	152.36	192.01	188.85	191.99	72.41	130.97
2	35.71	73.83	84.04	84.45	142.44	268.87	338.28
3	173.65	81.96	87.21	85.67	87.45	142.57	65.13
4	245.28	243.91	248.44	240.52	233.07	209.84	122.05
5	325.28	375.65	338.67	334.86	322.99	237	296.43
6	260.32	399.05	272.15	267.26	319.71	295.25	460.11
7	131.28	184.55	251.64	244.5	235.36	150.31	295.88
8	265.03	427.72	253.01	249.31	240.77	281.76	178

Table A.7: Mean TST for all methods in Environment 2, Distribution 2

A.2.2 Environment 2

The data in Table 6.5 in Section 6.5.2 is comprised of the Method with the lowest mean TST for each victim and distribution. Full data tables are shown here, with all mean TST values for all Methods shown for each distribution.

	Method						
Victim	0	1	2	3	4	5	6
1	22.08	46.43	83.09	100.49	91.48	17	16.97
2	59.52	292.31	262.64	253.37	289.76	364.61	438.21
3	122.8	49.44	86.16	85.39	102.01	90.77	68.47
4	176.79	85.61	86.4	85.47	141.53	150.04	60.91
5	39.57	28.09	23.32	23.28	24.16	24.71	17.87
6	241.77	227.69	244.04	248.27	246.24	221.28	120.44
7	241.77	318.92	244.04	248.27	246.24	221.28	123.29
8	248.64	522.07	256.96	258.03	255.28	300.63	168.72

Table A.8: Mean TST for all methods in Environment 2, Distribution 3

	Method						
Victim	0	1	2	3	4	5	6
1	229.87	134.24	190.4	188.23	210.55	68.88	128.88
2	231.95	235.4	190.41	189.77	210.55	110.04	173.19
3	39.25	84.56	85.95	86.76	32.25	24.04	222.43
4	25.51	86.53	90.23	131.31	70.09	15.03	17.16
5	57.95	158.79	145.76	160.59	111.73	285.8	378.68
6	218.81	147.59	236.73	236.89	237.39	165.03	276.23
7	232.77	325.57	257.21	264.2	265.28	241.8	371.19
8	107.17	134.09	153.83	154.32	143.71	304.41	388.19

Table A.9: Mean TST for all methods in Environment 2, Distribution 4

	Method						
Victim	0	1	2	3	4	5	6
1	181.85	85.76	86.13	86.24	87.52	128.91	63.28
2	245.08	210.96	238.67	238.89	215.29	194.11	122.55
3	50.05	85.76	86.01	86.13	87.08	48.95	37.4
4	344.29	305.47	327.69	323.93	341.73	227.09	299.95
5	131.16	74.39	91.01	94.24	133.97	36.17	241.75
6	264.48	378.73	276.83	276.37	306.53	243.61	462.48
7	226.88	320.48	272.18	261.79	278.29	230.05	388.99
8	222.57	228.35	270.73	257.59	280.07	237.52	388.99

Table A.10: Mean TST for all methods in Environment 2, Distribution 5

	Method						
Victim	0	1	2	3	4	5	6
1	206.52	209.71	176.32	201.66	201.4	208.87	129.12
2	206.51	88.25	176.31	200.35	201.4	208.87	129.12
3	248.44	337.76	225.55	239.81	242.08	208.97	228.84
4	221.04	216.89	223.96	215.32	221.31	190.15	179.03
5	221.04	191.49	224	215.32	221.27	190.15	179.95
6	145.2	186.16	144	138.45	140.07	117	263.37
7	140.09	111.33	143.75	138.25	139.97	203.45	329.05
8	145.2	111.49	143.99	138.45	140.07	115.37	263.99

Table A.11: Mean TST for all methods in Environment 3, Distribution 1

	Method						
Victim	0	1	2	3	4	5	6
1	201.52	191.36	180.73	194.26	195.95	208.44	135.68
2	201.52	133.69	182.27	193.08	195.95	208.77	135.68
3	139.87	91.71	143.15	139.88	140.89	188.77	365.29
4	219.19	98.23	223.84	219.03	223.09	189.31	214.44
5	219.19	105.84	223.84	219.03	223.09	189.31	214.44
6	139.87	86.33	143.13	139.88	140.88	199.27	336.47
7	139.87	262.79	143.15	139.88	140.89	199.84	371.89
8	143.36	86.35	143.27	140.43	140.89	113.59	299.2

Table A.12: Mean TST for all methods in Environment 3, Distribution 2

A.2.3 Environment 3

The data in Table 6.7 in Section 6.5.3 is comprised of the Method with the lowest mean TST for each victim and distribution. Full data tables are shown here, with all mean TST values for all Methods shown for each distribution.

	Method						
Victim	0	1	2	3	4	5	6
1	140.53	92.56	142.01	139.25	140.04	196.32	333.07
2	248.23	169.05	228.19	227.37	242.64	210.05	254.97
3	248.23	256.95	228.2	227.37	242.65	210.13	255.21
4	248.24	332.52	228.2	227.48	242.65	210.6	255.39
5	142.45	166.53	142.17	139.28	140.2	112.61	259.05
6	142.45	233.6	142.17	139.28	140.2	112.61	259.37
7	142.45	161.11	142.17	139.28	140.2	112.61	259.05
8	220.31	157.17	223.12	216.95	220.19	186.47	176.39

Table A.13: Mean TST for all methods in Environment 3, Distribution 3

	Method						
Victim	0	1	2	3	4	5	6
1	139.59	172.19	142.28	137.81	137.88	196.15	336.96
2	195.61	108.96	177.48	187.99	198.57	204.96	127.16
3	138.37	70.88	142.28	137.8	137.88	196.15	289.49
4	236.21	269.71	225.85	230.49	241.08	205.39	230.79
5	143.53	115	142.49	138.4	138.79	115.16	264.72
6	143.53	145.35	142.49	138.4	138.77	115.16	265.35
7	215.93	206.72	223.37	215.76	219.79	186.49	183.31
8	215.93	174.36	223.37	215.76	219.79	186.49	183.37

Table A.14: Mean TST for all methods in Environment 3, Distribution 4

	Method						
Victim	0	1	2	3	4	5	6
1	195.69	131.17	170.21	183.47	183.64	214.2	152.44
2	195.69	107.97	170.23	183.53	183.65	209.76	152.45
3	140.19	110.53	141.03	139.13	141.08	198.77	380.03
4	239.52	311.37	230.08	232.93	237.85	214.72	270.03
5	148.16	224.41	142.71	139.83	141.45	116.61	271.96
6	148.16	108.43	142.69	142.47	141.53	116.61	272.04
7	223.27	363.72	225.03	219.26	223.19	189.17	187.28
8	223.24	105.49	225.01	219.11	223.19	189.17	187.16

Table A.15: Mean TST for all methods in Environment 3, Distribution 5