

AN URBAN WARFARE APPLICATION OF SYSTEMS ENGINEERING FOR THE FIRST DERIVATIVE

Sonja Demuth
John Felini
Joseph Gerloff
James Mai
Nicholas Swingle
Phil Harton
Barry Horowitz

University of Virginia
Department of Systems and Information Engineering
Charlottesville, VA 22904

ABSTRACT

In an effort related to urban operations, the United States Army has funded a research project for the University of Virginia (UVA) that includes the design and development of a reconfigurable information management system (RIMS) mission-specific deployments of unattended sensors. Using a highly flexible networking technology called HyperCast, the RIMS project hopes to provide information in a dynamic and adaptable fashion over disparate networks (Liebeherr 1999). As an early step in the RIMS project, an experiment was performed to evaluate the prototype system's ability to be reconfigured and setup in different scenarios. By testing the time to configure and install the prototype, this experiment helps indicate the current system's ability to rapidly adjust. These results will act as an initial indication that the application of the reconfigurable concept can enhance the ability to rapidly set-up systems on a mission-specific basis.

1 INTRODUCTION

A significant number of military operations occur in a populated urban environments. In order for the military to be a lethal force in these urban operations, it must adapt to the changing environment while maintaining information superiority and situational awareness. Information superiority is access to information that is denied to the enemy while situational awareness is knowledge of one's own surroundings. The military also needs to be able to quickly react to new information in order to maximize the number of successful operations. These requirements give rise to a demand for mission support technology that can adapt to changing needs.

One solution to this demand is reconfigurable systems. The University of Virginia has produced a systems concept called "Systems Engineering for the First Derivative." This concept addresses how to create a rapid, functionally reconfigurable system that can provide the capabilities

needed by the user. HyperCast, also developed at UVA, is a technology that enables significant capabilities related to reconfiguration. HyperCast is a software protocol that builds and maintains overlay networks that allows information to be exchanged between specific groups. Each overlay network is designed such that mission-critical information is sent to only those users who need to know it. The technology supports large numbers of simultaneous groups to co-exist, so divisions of information exchange can be based on various groupings, such as job dependent, geography dependent, and security dependent. This project investigated how HyperCast, as a reconfigurable system, can meet the needs of the military when integrated with other military technologies.

The remainder of the report will discuss the outcomes of this project. Section 2 will discuss in further detail how the first derivative concept can be applied to urban warfare. Section 3 will address the technologies integrated with HyperCast to create a reconfigurable system. Section 4 will focus on the difficulties when designing the systems to fit certain scenarios. Section 5 will discuss the methodology of an experiment conducted to evaluate the difficulties in reconfiguring the system. Section 6 will review the results collected from this experiment. Section 7 will end with an interpretation of the project and recommendations for future research that could improve the system.

2 THE APPLICATION: URBAN WARFARE

While the first derivative concept is applicable to a variety of situations, this paper focuses on its application to urban warfare. Urban warfare can be thought of as having two components: terrain and operations. Both components present obstacles and complications for military actions in or near cities.

2.1 Urban Terrain

One of the main complexities of urban warfare is the terrain itself. Fundamentally, urban warfare is fought in or near a city. As a result, military operations are affected by the dynamic qualities of a city, with its continuous flow of people, goods, services, and information into, from, and within its walls (United States Joint Chiefs of Staff 2002). In fact, urban areas are regarded as the most complex physical terrain.

While urban areas can vary greatly, they generally have the following three characteristics: “a complex man-made physical terrain, ...a population of significant size and density, ...and an infrastructure upon which the area depends...”(United States Joint Chiefs of Staff 2002). The differentiator between urban scenarios and other operations is the “impact of military operations on the urban population and vice versa” . The military must be concerned not only with its enemy, but also the safety of the surrounding civilian population. In addition to the urban population, difficulties arise due to the presence of buildings and sub- and super-surface areas. Not only does the variety in structures and levels of terrain hinder visibility and the ability to gather data, but it also increases the difficulty of communication across the diverse terrain. Soldiers become segregated from building to building, subsurface to supersurface, increasing the likelihood that important information is either lost in the translation or simply unable to be transferred (see Figure 2).

2.2 Urban Operations

Like the terrain of urban areas, the operations themselves are often highly complex, as well as being different from one case to the next. However, urban operations do have some common characteristics. Quite ideal for an uneven battlefield, urban areas allow smaller, less sophisticated forces to battle otherwise superior opponents. The urban terrain, with many buildings and subsurface areas, can easily diminish any technological advantages (United States Joint Chiefs of Staff 2002). Even if the location of a target is narrowed down to an urban region, finding and neutralizing that enemy can still be like finding a needle in a haystack. In addition, the need to maintain infrastructures and the security of civilians greatly increases the constraints placed on military operations. Whether it means using less effective weaponry that is more precise or not engaging a target due to surrounding civilians, the Army can be handcuffed by the need to minimize collateral damage in urban settings. Another issue the Army faces is that urban operations are often weakened by becoming decentralized. As soldiers disperse to buildings and underground areas, communications and control are taken out of the hands of commanders and put on the shoulders of smaller units.

COMPARISON OF OPERATIONS IN URBAN AREAS AND OTHER TYPES OF ENVIRONMENTS				
CHARACTERISTIC	URBAN	DESERT	JUNGLE	MOUNTAIN
Number of Noncombatants	High	Low	Low	Low
Amount of Valuable Infrastructure	High	Low	Low	Low
Multi-dimensional Battlespace	Yes	No	Some	Yes
Restrictive Rules of Engagement	Yes	No	No	No
Detection, Observation, Engagement Ranges	Short	Long	Short	Medium
Avenues of Approach	Many	Many	Few	Few
Freedom of Vehicular Movement and Maneuver	Low	High	Low	Medium
Communications Functionality	Degraded	Fully Capable	Degraded	Degraded
Logistics Requirements	High	High	High	Medium

Figure 1: Comparison of urban versus other operational environments (U.S. Joint Chiefs of Staff 2002)

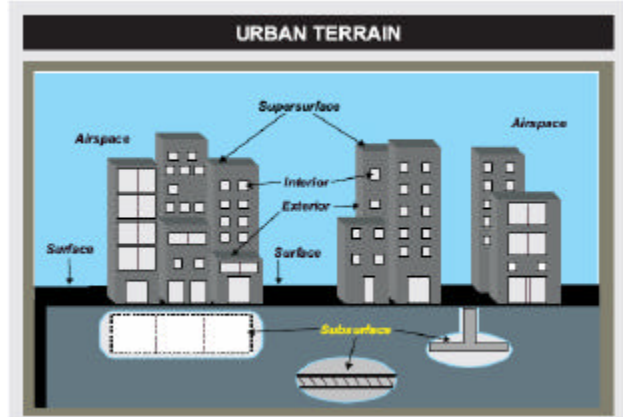


Figure 2: A visual breakdown of urban terrain (U.S. Joint Chiefs of Staff 2004)

2.3 Urban Warfare and the First Derivative

The first derivative could greatly help not only decentralization, but also many of the other obstacles and complexities associated with urban terrain and operations. The dynamic and complex qualities of urban warfare are the ideal situation for a highly reconfigurable information system. Furthermore, the ability to communicate in an ad-hoc fashion across disparate networks is badly needed to help the military overcome decentralized and disconnected communications.

2.4 HyperCast

In order to enhance the military’s information systems, the first derivative project plans to use a technology named HyperCast as its backbone. HyperCast is “a distributed ad hoc network management technology consisting of a suite of network management protocols and corresponding software” (Horowitz 2004). The first derivative has a need for a peer-to-peer (P2P) network technology, with P2P being defined as “a network in which each workstation has equivalent capabilities and responsibilities” (Kearns 2003).

In terms of the project for this paper, HyperCast handles distribution of data throughout the information man-

agement system, such as transfer of sensor data from a laptop located in a hostile area to soldiers patrolling nearby with PDAs to other computers back at headquarters. HyperCast can be thought of as a chat room, in a way. Multiple users can sign into or out of the room, as well as send and receive messages to either all members in the chat room or a select sub-group of the chat room. HyperCast handles building the chat room, or overlay network, that provides an environment for users to join, exchange data, and eventually exit.

HyperCast is technically-defined as a software that “builds and maintains logical overlay networks between applications, and supports data transmission between applications in the overlay” (HyperCast Team 2001). In regards to the Internet, an overlay network “is a virtual network that is implemented on top of a network of routers and links” (HyperCast Team 2001). A HyperCast overlay network is a logical network, allowing the transfer of data amongst nodes on the network without concern for the underlying physical network connecting the nodes. “Nodes in the overlay network can be hosts, routers, servers, or applications”.(HyperCast Team 2001) HyperCast is capable of using different topologies, or methods of building its overlay. Regardless of the topology, all data are transmitted along spanning trees with each node communicating with its neighbors. Communication between nodes can be specified to be either unicast or multicast. Unicast communication is aimed at getting a transmission forwarded along the shortest path to the root node of the tree. On the other hand, multicast indicates that the transmission is downstream from the root node to all of the tree’s children nodes.

Not only does HyperCast handle moving data from node to node, the software also provides an interface for applications to send and receive the transferring data (Liebeherr, Wang, and Zhang 2001). The means for programs to interact with the overlay network are socket overlays. A socket overlay is an application programming interface (API) that allows programs to utilize HyperCast overlay networks, but at the same time ignore the details of the physical network and the overlay network topology. Each overlay socket has a logical address that keeps its place in the logical overlay network and a physical address necessary for actually transferring the data on the physical network. Application programs using the socket overlay only see the logical address. Basically, an application programmer can send or receive data from an overlay network without any consideration for how the data is actually getting to the node hosting his application. By abstracting the transfer of data, HyperCast makes it easy to interface with the overlay network while also maintaining high flexibility

in overlay topologies and physical overlays.

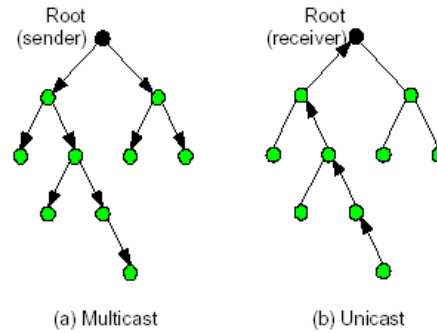


Figure 4: Data forwarding in overlay networks (Liebeherr, Wang and Zhang 2001)

While HyperCast has not been previously used for military applications, it has been researched in the realm of emergency response, an area similar to military urban scenarios (Kearns 2003). Like urban military operations, emergency response situations are often highly complex and time critical. One of the key features of HyperCast that was found to be useful for emergency response was the software’s ability to form the overlay network very quickly. Not only did HyperCast quickly create the overlay, but it also adjusted quickly and easily when nodes, or members, were added or subtracted from the network. HyperCast even allowed for nodes to be members of multiple overlays, yet another feature that benefited the emergency response project in the past and is ideal for the first derivative.

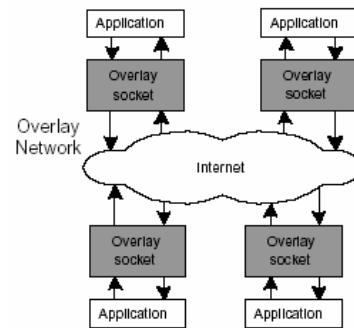


Figure 5: The overlay network is a collection of overlay sockets (Liebeherr, Wang and Zhang 2001)

3 ENABLING TECHNOLOGIES

In addition to consumer-grade laptops and PDAs, the re-configurable system developed through this endeavor employs four major hardware components: the Stargate[®] computer, the NovaRoam[®] EH900[™] Mobile Router, the MICAz[®] processor/radio platform, and three types of sensors. Each is discussed below.

The Stargate is a small, single board computer that typically runs a lightweight version of Linux. It allows for highly reconfigurable systems as it can interface with MICAz units and USB devices, can communicate via ZigBee (through attached MICAz), 802.11, and wired Ethernet, and can run many applications for the Linux platform, including the Java Virtual Machine. This activity demonstrates the versatility of the Stargate through its roles involving imagery, sensor data collection, early warning decision logic, and bridging ZigBee and 802.11 networks, all of which are described in Section 4.

The NovaRoam EH900™ is a wireless router with a range of up to 25 miles (line of sight) and a 100BT Ethernet interface. It has a simple yet robust web-based configuration interface making it easy to incorporate into networks, though in most cases no special configuration is needed. A group of EH900s in range of each other can automatically form a network which appears to the attached devices as a wired, switched network. Its range, transparency, and ad-hoc capabilities make it ideal for rapidly reconfigurable systems in areas where connectivity is difficult. The NovaRoam EH900 can also be DC powered in an automobile, allowing its users to be mobile.

The MICAz is a tiny computing platform with built in wireless communication through the ZigBee (802.15.4) protocol. It is designed specifically for deeply embedded sensor networks. It is ideal for reconfigurable systems because it is a reprogrammable device suitable for gathering and transmitting information. In this activity, the MICAz is the information provider of the system. It gathers data directly from sensors and transmits it, potentially through other MICAz's via multi-hop routing, to MICAz's attached to Stargate gateways through a serial interface.

Three types of sensors are used in this activity: magnetic, seismic, and passive infrared. Magnetic sensors detect changes in the magnetic field, for example due to a vehicle passing near the sensor. Seismic sensors detect vibrations in the ground and can be set to different sensitivities. Passive infrared sensors detect changes in infrared radiation, reliably detecting movement of humans across a line of sight. Each sensor has a standard connector that can be attached to a MICAz unit in combinations of up to 3 sensors. Used together they can provide more information than any one sensor, allowing them, for example, to discern between people and vehicles. This modularity allows a system to be easily reconfigured for various sensory needs.

4 SCENARIOS

Two different applications for unattended sensors have been used to demonstrate the full functionality of the described reconfigurable systems. The following sections discuss the cases in detail. In one case, called the Man In

The Loop (MITL) scenario, if intruders approach a restricted area and are detected in a warning zone, the surveillance system switches into a more sensitive state of high alert. Should the intruders get closer to the restricted area, cameras photograph them and sensors alert security personnel and report on their location. This scenario also included a Talon Robot, which is a military robot equipped with five mounted cameras and a weapon attachment. The robot can be called upon to intercept the intruders, if desired. In the second case, called the Distributed Sensor (DS) scenario, should intruders enter a monitored building, nearby officers receive up-to-date information on the intruder via their PDA. Should an officer go to the building with the intruders, authentication via a digital certificate is required so that video images can be received about the intruders.

4.1 MITL Scenario

The focus of this scenario was on monitoring an area with restricted access. To do this, three monitored zones on the UVA grounds were used, one Early Warning (EW) zone and two Area Denial (AD) zones, as shown in Figure 6. The EW zone was by an entrance gate and the AD zones were located near the front and back doors of a building. Each zone used a combination of motion, seismic, and magnetic sensors to detect intruders passing through the zones. The combination of readings from each of the different sensors allowed users to distinguish between people and vehicles. The AD zones also had cameras to take photographs when its sensors detected an intruder. The sensors in the AD zones had two different modes, passive and active, which determine the logic for combining sensor detections. Two different sensor detections were required for what is considered to be a detection of an intruder in passive mode. When the EW zone sends a sensor alert, the logic for sensor integration was adjusted to any single sensor detection. In addition, there was a base station laptop, which received sensor trigger alerts from all zones, as well as images from the AD zones. The Talon Robot operator station was positioned near the AD zones and received sensor alerts and images from them. If the station received an alert, the robot could function as a mobile unit to further investigate the AD zones.

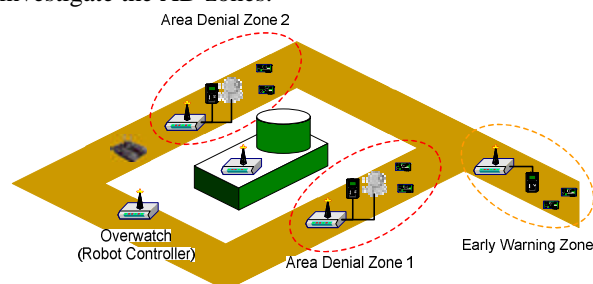


Figure 6: Scenario 1 Physical Layout (Sung 2005)

The scenario for our information management system to support the MITL configuration works in the following manner: an intruder vehicle passed through the EW zone and triggered the combination sensors. The Gateway in the EW zone sends a signal to the base station as well as the AD zones to switch to active monitoring mode. The intruder then exits the car and proceeds on foot to show that the combination sensors can differentiate between vehicles and people. When he enters AD 1 the sensors detect him, which triggers the camera to take images of that zone. The Gateway in AD 1 sends the alert messages and images to the base station. Then, the operator deploys the robot into AD 1, which will send its video imagery back to the operator station. The images are forwarded to the base station and the base station commander sends a “disable intruder” command to the robot operator. The robot operator then “fires” the weapon, which is simply a mounted web cam that would send images back with crosshairs on the target.

4.2 DS Scenario

In the second system configuration, the main scenario focus is on patrolling officers being able to gain access to updated security information as they approach a building. For this scenario, the project used two monitored zones inside a building, each different room simulating a zone in separate buildings, as shown in Figure 7. The monitored zones used motion sensors and cameras to detect personnel entering or leaving the zone, as well as take pictures of them. The system sent the information to different users with different levels of detail and latency depending on the user’s mission. Figure 7 shows this with the Roving Patrol outside of the building, who enters both Zones 1 and 2, and receives updated information on his PDA for each zone as he enters them. The patrol gains access to the information in each zone by sending an access request message to the base station. The access request message includes a digital certificate which will allow the base station to determine if the patrol has the correct authorization and access the information.

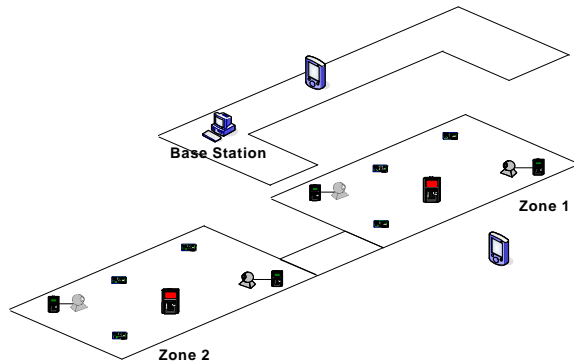


Figure 7: Scenario 2 Physical Layout (Sung 2005)

The demonstration for this information management system works in the following manner: an intruder enters

Zone 2, triggering a motion sensor which causes the Zone 2 Gateway to signal the camera to capture images. The intruder then leaves Zone 2 and the Gateway sends sensor alert messages and images to the base station. A roving patrol with a PDA approaches Zone 2 and sends an access request message to the base station. The base station authorizes the patrol, allows him to join the Zone 2 Overlay, and sends him the summary update of Zone 2. The intruder then enters Zone 1, again triggering the sensors and cameras in that zone. Sensor alert messages and images from Zone 1 are then sent to the base station. The roving patrol sends an access request message and joins the Zone 1 Overlay in a similar manner. After the roving patrol receives the summary update from Zone 1, he hurries over to Zone 1, exiting the Zone 2 Overlay in the process. Once in Zone 1, the roving patrol receives live sensor and image information of that zone, allowing him to apprehend the intruder.

4.3 Design and Implementation Difficulties

When configuring the software and hardware for each scenario, there are several potential difficulties to be encountered in both the design and implementation of the system. One problem can be caused by the lack of technical expertise of the individual setting up the system. It was assumed that these individuals would have limited knowledge of HyperCast. These soldiers must be able to dynamically create a system design adapted to their specific environment. This design must employ proper overlays that establish information exchange groups with supporting communication between the group members, along with ensuring that parts are within communication range. This person responsible for the system configuration must also be able to set up the software and hardware properly, and test the system to ensure functionality. Finally, proper maintenance of the system must be provided, including a critical requirement to frequently monitoring batteries and being on call in case of device failures.

4.4 Conceptual Design

The utility of the HyperCast system depends primarily on the layout of integrated hardware and communications between them. The layout is based on both the data to be collected and the recipients of this data. The HyperCast overlays should link a data source, such as cameras and sensors, with the desired recipient. For example, in the MITL scenario, the Early Warning overlay links the sensor in the Early Warning zone to the base station. These overlays will allow for efficient communications between these applications; however the configuration must ensure that parts are within communications range. The hardware depends on communication protocols provided by wireless 802.11 Ethernet and Nova-Roam data radios. Depending

on the location of the data source along with the proximity of the related receiving node, the developer must employ the proper wireless devices to ensure reliable communication between parts.

4.5 Implementation Problems

With the proper layout decided upon, the system developers must now implement the layout using the issued software and hardware. HyperCast makes the software and network setup very simple, especially if the overlays have already been decided upon in the conceptual stages. Nodes must be created in proper overlays and message types must be defined according to the information traveling between parts. Finally, the hardware must be setup according to the designated layout. This involves proper placement of sensors, cameras, gateway computers, NovaRoam radios, and base station.

The major obstacles faced during the design phases involved actual programming on the MICAz. This programming dealt with both receiving sensor information from the MICAz along with sending camera images to the base station. These obstacles however lie only in the design phase; reconfiguration will not involve this programming on the MICAz. Configuration efforts would know exactly how to interface the individual parts and integrate the parts using the wireless communications. However, one final implementation step involves testing the system. Depending on the characteristics of the intruders under surveillance, sensor sensitivities and algorithms must be established that provide a proper tradeoff between missed events and false alarms. System configuration will depend on this information, since each instance of the system may be designed to track different intruders (such as single soldiers, troops, cars, tanks, etc.).

4.6 System Maintenance

After the system layout and design is complete, the final application of the HyperCast system involves hardware maintenance. The utility of the system relies on the persistent monitoring of parts, since device failure causes complete system shutdown. The overall system must be monitored to make sure that battery levels are always sufficient in the individual parts. The Stargate computers also can crash, and the individual motes occasionally burnout. Assigned personnel must be ready for such failures and be quick to replace or fix malfunctioning hardware. The initial system configuration should also include redundancy to account for possible failures.

5 EXPERIMENT

The objective of the experiment was to gauge the time required for non-experts to set up an unattended grounded

sensor scenario, such as ones discussed in the previous section. It is possible to modify a configuration by customizing only minor components and reusing the architectural services provided by Hypercast. This experiment only dealt with software configuration and it was assumed that all hardware was available and needed no modifications. The reconfiguration of the technologies used was examined through setting up the DS scenario by 5 subjects who are non-expert in computer programming and network engineering. The subjects were first trained in the basic tasks required by reconfiguration outside the context of a particular scenario. Other assumptions included that the subject had the setup instructions, all batteries are charged, and the motes are preconfigured. The procedure is provided in Appendix 1. All the applications that were needed to be applied to the different system components were stored on a USB memory stick. There were four steps in completing the experiment: (1)laptop configuration, (2) Stargate configuration, (3) PDA configuration, and (4) physical placement of equipment. The time in terms of hour and minutes was recorded after each step was completed.

The laptop was viewed as the base station in the experiment. Configuring the laptop entailed configuring the network interfaces and copying and running an application from a USB memory stick. Configuring the network interfaces required the subject to set the IP addresses of the laptop and sign on to the "tsunami" wireless network. A folder on the memory stick was then saved onto the laptop. Once the folder was located in the C drive, the subject used the command prompt to execute the application.

Once this application was running, there were six Stargates that needed to be configured. The Stargates were connected to the laptop via serial cable and to the USB memory stick. The subject used a terminal application to move the application from the memory stick to the Stargate and then execute it. For the Stargates with cameras, the application called "camera_node" was executed while the other Stargates executed the application called "gateway".

Once all six Stargates completed this process, two PDAs need to be configured. Configuring the PDA was similar to configuring the laptop. First the subject needs to check to assure that the wireless interface is turned on. Then the network settings need to be configured. The PDA, like the laptop, needs to be connected to the "tsunami" wireless network. Once the network is set up correctly, the DS_PDA directory needs to be moved from the memory stick to the PDA. Because this directory can not be moved to the PDA directly, a partnership is established with the PDA and laptop. Through an application called Microsoft ActiveSync, any folder from the laptop can be saved onto the PDA. Thus, through this application, the DS_PDA folder is saved onto the PDA.

Once all the equipment is configured, the subject needs to arrange the equipment as arranged in Figure 8.

There are two zones that contain the Stargates, motes, and cameras. The base station (the laptop) is outside of this zone as are the two PDAs. Once this process was completed, the elapsed time and any problems that occurred were recorded.

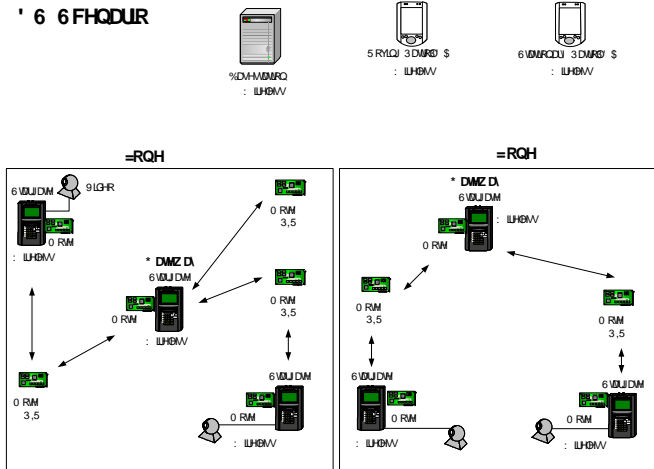


Figure 8: DS Scenario Set Up (Sung 2005)

6 RESULTS

The results from the experiment are shown in Figure 9. There are a few numbers worth highlighting: on average, the reconfiguration took about 76 minutes. Across all trials, the majority of the time was spent on the tasks of Section 2 (Stargate configuration), with a maximum time of 83 minutes and a minimum of 35 minutes in that section. The least amount of time was spent in Section 4 (Physical Setup), on average under 3 minutes. The box plots in Figure 10 display the maximum and minimum times for each section, as well as the medians and any outliers. As was expected from the raw numbers, Figure 10 shows visually that Section 2 more than doubles the times of all of the other sections combined.

	Sec. 1	Sec. 2	Sec. 3	Sec. 4	Total
Trial 1	9	48	10	2	69
Trial 2	5	83	10	2	100
Trial 3	3	35	7	3	48
Trial 4	6	57	13	2	78
Trial 5	9	56	14	5	84
Avg	6.4	55.8	10.8	2.8	75.8

Figure 9: Resulting times (in minutes) for the experiment

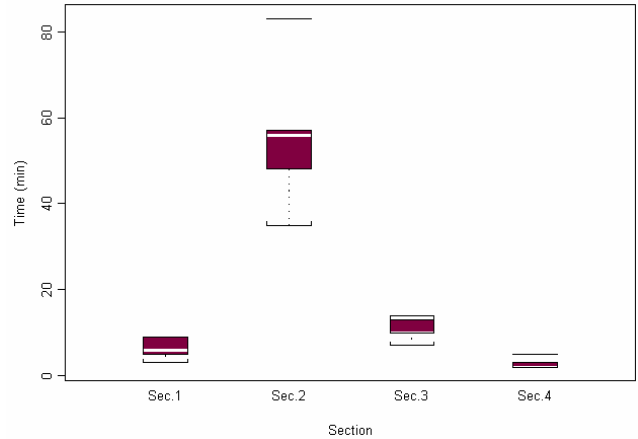


Figure 10: Box plots of the experiment results

7 DISCUSSION

7.1 Interpretation of results

While it is hard to draw conclusions from such a small sample set, some observations can be made from the experiment's data. For instance, only one subject managed to reconfigure the system in under an hour. Sections 1 (Laptop configuration) and 4 appear to only nominally affect the reconfiguration time. The bottleneck of reconfiguration, as can be seen in Figure 10, clearly lies currently in the tasks of Section 2. One possible reason for the bottleneck are slow transfer rates between the Stargate computers and the USB memory stick; depending on the file size and disk space available on the Stargate, the transfer took up to 2 minutes for the camera nodes and 5 minutes for the gateways, of which there are 4 and 2 respectively. In addition, the interface and programmatic commands needed to interact with the Stargates were most likely the least familiar and most complex functions for each of the five subjects in the experiment. Along the same lines, subject 3 had previous experience with owning a PDA, which may account for the minimum time in Section 3 (PDA configuration). Whatever the cause may be, it appears there exists some way to improve the reconfiguration time in Section 3 and may warrant further investigation. Similar minor reductions in time may be possible in Section 1 as well. However, Section 2, which on average accounted for nearly 74% of the total time, clearly warrants the most attention for future improvements.

7.2 Difficulties Encountered

The most significant problem encountered during the experiment was the difference in domain knowledge between the author of the procedure and the actual subjects. Many functions viewed as intuitive by the author proved to be difficult to the subjects because of a lack in procedure de-

tail. For instance, much of the procedure involved using Linux commands to interface the individual parts. While the author of the instructions is experienced with this operating system and actually wrote part of the program code during system development, the actual subjects had no experience with Linux and were not able to easily understand and issue commands. Had the author detailed not only user input but also the proper system feedback, subjects could have better evaluated their position within the procedure and also their individual mistakes.

The author's intuition assumption not only was a barrier to the software programming but also to hardware setup. Many repeated steps, such as the insertion of a memory card, were mentioned only the first time they were used. While the author assumed the subjects would know to repeat these steps for subsequent hardware units, the subjects were confused by the lack of detail. This problem yielded inflated experiment times, since the subjects had to spend ample time detecting the missing steps and then accounting for them accordingly.

The final difficulty encountered during testing was the lack of a controlled environment. Many of the hardware units used for the experiment were concurrently being used by other group members for further development. These other members manipulated many of the units so that their starting status was different than the author of the instructions had assumed. The manipulated hardware could not be used for testing until it was returned to its original status, which greatly hindered the progress of the experiment.

7.3 Implications for Future Development

The purpose of this experiment was to get a sense of the time taken and difficulties encountered for reconfiguration of the system by non-experts. The average times for configuration of the laptop, configuration of the PDAs, and physical placement of the equipment was relatively small, with relatively low deviation between subjects. By comparison, both the average and range of times for Stargate configuration was very high. As discussed previously, lack of explicitness and detail in the procedure coupled with unfamiliarity on the part of the subjects was the main source of difficulty. These factors, as well as the lack of a controlled environment, were external to the design of the system itself, but can be alleviated through improvements to the system directed at ease of reconfiguration.

The most straightforward way to deal with users' lack of familiarity with the components is to have them perform fewer steps. This can be achieved through automation: for example, the most difficult and time consuming step in the experiment – copying and running software on the Stargates – could have been done by delivering the software via wireless networking from a distribution server and performing file operations and program execution locally

through batch scripts. This approach would have allowed all 6 Stargates to be reconfigured simultaneously, where the current procedure relies on both the laptop and memory stick, forcing the operation to be serial.

To address a potentially uncontrolled environment, and more importantly to provide a better level of assurance, automation of procedure should be complemented by automated state checking and testing of the system. Ideally, the states of each component, including failure modes, could be determined beforehand and addressed individually in the automated reconfiguration software. The state of the system would be a combination of the states of the individual components, and the reconfiguration software would contain mappings between tests and states, and states and corrective actions. However, with complex systems, it is difficult to enumerate all of the possible states, let alone design tests to identify each state with confidence. Future work will address these issues to make field reconfiguration of complex systems more viable.

ACKNOWLEDGMENTS

We would like to acknowledge the SEAS Army-HyperCast Team for their efforts in researching and developing these systems. We could also like to thank the United States Army for giving us the opportunity to work with them.

APPENDIX: DS SCENARIO INSTRUCTIONS

Disposable Sensor Configuration Experiment

Objective

To gauge the time required for non-experts to set up an unattended ground sensor scenario

Procedure

1. Laptop configuration
 - a. Record the current time
 - b. Configure network interfaces
 - i. Right-click the wireless networking icon in the system tray on the lower-right portion of the screen. Click "View Available Wireless Networks"
 - ii. Click "Change the order of preferred networks" on the left
 - iii. Click Advanced
 - iv. Make sure "Access point (infrastructure) networks only" is selected and click Close
 - v. Click the General tab at the top left of the window
 - vi. Click "Internet Protocol (TCP/IP)," click Properties.

- vii. Click the radio button for “Use the following IP address:”
 - 1. Enter 192.168.10.1 for the IP address
 - 2. Enter 255.255.255.0 for the subnet mask
 - 3. Leave Default gateway and DNS servers blank, click OK
 - viii. Click OK again to close the “Wireless Network Connection” window
 - ix. Right-click the wireless networking icon in the system tray on the lower-right portion of the screen. Click “View Available Wireless Networks”
 - x. Select “tsunami” and click connect at the bottom-right
 - xi. Close the window
 - c. Copy DS_BaseStation directory from memory stick to C:\ on laptop (see 4.1)
 - d. Start the application by running C:\DS_BaseStation\basestation.bat
 - e. Shut down the application
 - f. Delete C:\DS_BaseStation
 - g. *Record the current time*
2. Stargate configuration
- a. *Record the current time*
 - b. Log into the Zone 1 gateway (see section 4.4)
 - c. Type cp
/mnt/usb1/DS_Stargate/gateway.tar /
 - d. Type rm /gateway.tar
 - e. Type cd /root
 - f. Type ./start_asf &
 - g. Type cd hypercast_test
 - h. Type start_gateway_zone1 &
 - i. Leave HyperTerminal running and unplug all cables from the Zone 1 gateway and repeat steps b-f for the Zone 2 gateway
 - j. Type start_gateway_zone2 &
 - k. Log into one of the camera nodes
 - l. Type cp
/mnt/usb1/DS_Stargate/camera_node.tar /
 - m. Type rm / camera_node.tar
 - n. Type cd /root
 - o. Type ./start_asf &
 - p. Type start_source &
 - q. Leave HyperTerminal running and unplug all cables from the camera node and repeat steps g-j for the rest of the camera nodes
 - r. Close HyperTerminal and do not save the connection when asked
- s. *Record the current time*
3. PDA configuration
- a. *Record the current time*
 - b. Make sure the PDA’s wireless interface is turned on
 - i. Turn on the PDA using the circular button in the top-right corner
 - ii. Tap Start
 - iii. Tap “iPAQ Wireless”
 - iv. Ensure that the WLAN icon is green. If not, tap it.
 - v. Click the X on the upper-right
 - c. Configure PDA network settings
 - i. Wait for the flashing light in the top-left corner of the PDA to turn green
 - ii. Tap the wireless icon next to the speaker icon at the top of the screen
 - iii. Tap Settings
 - iv. Tap Advanced
 - v. Tap Network Card
 - vi. Make sure the PDA is connected to tsunami. If not, tap and hold on tsunami and click Connect.
 - vii. Click ok on the upper right, then click it again
 - d. Copy the DS_PDA directory from the memory stick to a temporary directory on the laptop (you can drag the folder onto the desktop)
 - e. Copy the DS_PDA directory from the temporary folder on the laptop to the root directory of the PDA (see section 4.2)
 - f. Remove the PDA from the cradle
 - g. Locate the DS_PDA directory in File Explorer (see section 4.3)
 - h. Tap and hold the DS_PDA directory, then click delete and confirm
 - i. Turn off the PDA
 - j. Repeat steps b-i for the other PDA
 - k. *Record the current time*
4. Physical placement of equipment
- a. *Record the current time*
 - b. Place hardware as shown in Figure 1. Disregard the numbers, just make sure all the Zone 1 equipment is in zone 1, etc.
 - c. Place the Zone 1 gateway next to mote 36 in the picture
 - d. Place the Zone 2 gateway next to mote 45 in the picture
 - e. *Record the current time*
5. Return all equipment to its original location

REFERENCES

- Beam, T. K., & Liebeherr, J. 1999. HyperCast: A protocol for maintaining multicast group members in a logical hypercube technology. v 1736(Proceeding of the First International Workshop on Networked Group Communication (NGC '99), in: Lecture Notes in Computer Science) p. 72-89.
- Crossbow Technologies Inc. 2005. MICAz ZigBee Series (MPR 2400). [online] Available online via <xbow.com/Products/productsdetails.aspx?sid=101> [accessed April 4, 2005]
- Crossbow Technologies Inc. 2005. Stargate (SPB400). [online]. Available online via <xbow.com/Products/productsdetails.aspx?sid=85> [accessed April 4, 2005]
- Horowitz, B.M. Project Abstract [online]. Available online via <<http://www.sys.virginia.edu/capstone/2005/09.htm>>. [accessed September 13, 2004]
- HyperCast Team. Department of Computer Science, University of Virginia. HyperCast 2.0 Design Document. Available online via <http://www.cs.virginia.edu/~mngroup/hypercast/designdoc/Chp1-Overview/Chp1-Overview.html>>. [accessed October 25, 2004]
- Kearns, J. 2003. Application of Peer-to-Peer Technology to Advanced Emergency Response. p. v, 10, 34.
- Lorincz, K. 2001. Hypercast: A super-scalable many-to-many multicast protocol for distributed internet applications.
- Nova Engineering Inc. 2005. NovaRoam EH900™ Mobile Router. [online] Available online via <nova-roam.com/Inside.asp?n=Solutions&p=NR-EH900> [accessed April 4, 2005]
- Sung, Myong. February 2005. February Demo Overlay Design [internal document] p. 7.
- US Army. January 23, 2004. "5-3. FOC-05-03: Operations in urban and complex terrain".
- US Army. (2000). Army field training manual 3-06
- United States Joint Chiefs of Staff. 2002. "Doctrine for Joint Urban Operations" [online]. Available online via <<http://www.dtic.mil/doctrine/jel/new%5Fpubs/jp3%5F06.pdf>> p. 13-20.
- Wang, G., Zhang, W., Cao, G., & La Porta, T. 2003. On supporting distributed collaboration in sensor networks. Proceedings - IEEE Military Communications Conference MILCOM MILCOM 2003 - 2003 IEEE Military Communications Conference(v 2 2003) 752-757.
- She will be commissioned as a second lieutenant into the Air Force where she will be a Developmental Engineer at Los Angeles AFB. She can be contacted by e-mail at sonja@virginia.edu
- JOHN FELINI** attends the University of Virginia as an undergraduate pursuing a Bachelor's of Science in Systems and Information Engineering and a minor in Engineering Business. He will graduate in May and then pursue a career with Booz Allen Hamilton in Northern Virginia. He can be contacted at jfelini@virginia.edu.
- JOSEPH GERLOFF** is graduating in May 2005 with a degree in Systems Engineering with a minor in Engineering Business. He can be contacted at jag9q@virginia.edu.
- PHIL HARTON** has a BS in Computer Science from the University of Virginia and is currently pursuing an MS in Systems Engineering. He will begin working for Microsoft in Redmond, Washington starting in the summer of 2005. He can be contacted at plh5n@virginia.edu.
- DR. BARRY HOROWITZ** is a professor of Systems Engineering at the University of Virginia. He can be contacted at bh8e@virginia.edu.
- JIM MAI** is an undergraduate student at the University of Virginia majoring in Systems Engineering. In the summer of 2005 Jim will begin working for Northrop Grumman in northern Virginia. He can be contacted at jsm8b@virginia.edu.
- NICK SWINGLE** is an undergraduate student at the University of Virginia pursuing a Bachelor's of Science in Systems Engineering with a concentration in Computer and Information Systems. He can be contacted at nss5c@virginia.edu.

AUTHOR BIOGRAPHIES

SONJA DEMUTH is an undergraduate student at the University of Virginia pursuing a Bachelor's of Science in Systems Engineering with a concentration in Computer and Information Systems and a minor in Applied Mathe-