Ad-hoc Network Wireless Simulator
Concept of Operations

Version 1.1

Group G1: CJ Elliott, Greg Moser, James Woo, Matt Roberts and Yali Zhao

September 13th 2010

Hood College Computer Science Department

CS 524 Software Engineering

Version History:

1.1

- Cover page
- Document version
- Updated state diagram
- Revisions to sections 3, 4, 5, 6, and 8
- Glossary

1.0

Initial release

Abstract: The format and contents of a concept of operations (ConOps) document are described. A ConOps is a user-oriented document that describes system characteristics for a proposed system from the users' viewpoint. The ConOps document is used to communicate overall quantitative and qualitative system characteristics to the user, buyer, developer, and other organizational elements (for example, training, facilities, staffing, and maintenance). It is used to describe the user organization(s), mission(s), and organizational objectives from an integrated systems point of view.

Keywords: Ad Hoc, wireless, concept of operation, concepts of operations document, ConOps, developer, networking, operational requirements, scenario, simulator, user requirements, viewpoint

1.0 Scope

The mobile ad hoc network simulator's purpose is to provide a GUI demonstration of a wireless ad hoc network environment that exercises specific protocols using different scenarios and presents results in a user friendly format. The simulator's development will be based on the AODV Simulator [6] and constructed by students at Hood College's Computer Science Department. Once completed the simulator will be made available to the public for educational purposes and further research. The software will be contain an installer, installation guide, user guide and a programmer guide.

2.0 Referenced Documents

1. Magnus Frodigh, Per Johansson and Peter Larsson, Wireless ad hoc networking - The art of networking without a network, Ericsson Review No. 4, 2000, http://www.ericsson.com/ericsson/corpinfo/publications/review/2000_04/124.shtml
2. John Heidemann, Nirupama Bulusu, Jeremy Elson, Chalermek Intanagonwiwat, Kun-chan Lan, Ya Xu, Wei Ye, Deborah Estrin, Ramesh Govindan, Effects of Detail in Wireless Network Simulation, September 26th 2000, http://www.isi.edu/~johnh/PAPERS/Heidemann00d.html
3. Tony Larsson, Nicklas Hedman, Routing Protocols in Wireless Ad-hoc Networks - A Simulation Study, 1998:362 ISSN: 1402-1617 ISRN: LTU-EX--98/362--SE, http://epubl.luth.se/1402-1617/1998/362/index.html
4. Luke Klein-Berndt (NIST), A Quick Guide to AODV Routing, http://www.antd.nist.gov/wctg/aodv_kernel/A%20Quick%20Guide%20to%20AODV%20Routing.pdf
5. Charles E. Perkins (IBM, Watson Research Center), Pravin Bhagwat (Computer Science Department University of Maryland), Highly Dynamic Destination-Sequence Distance-Vector Routing (DSDV) for Mobile Computers, http://www.cs.virginia.edu/~cl7v/cs851-papers/dsdv-sigcomm94.pdf
6. Ali Dabirmoghaddam, Masoud Moshref, AODV Simulator, http://aodvsimulator.sourceforge.net

3.0 Current System or Situation

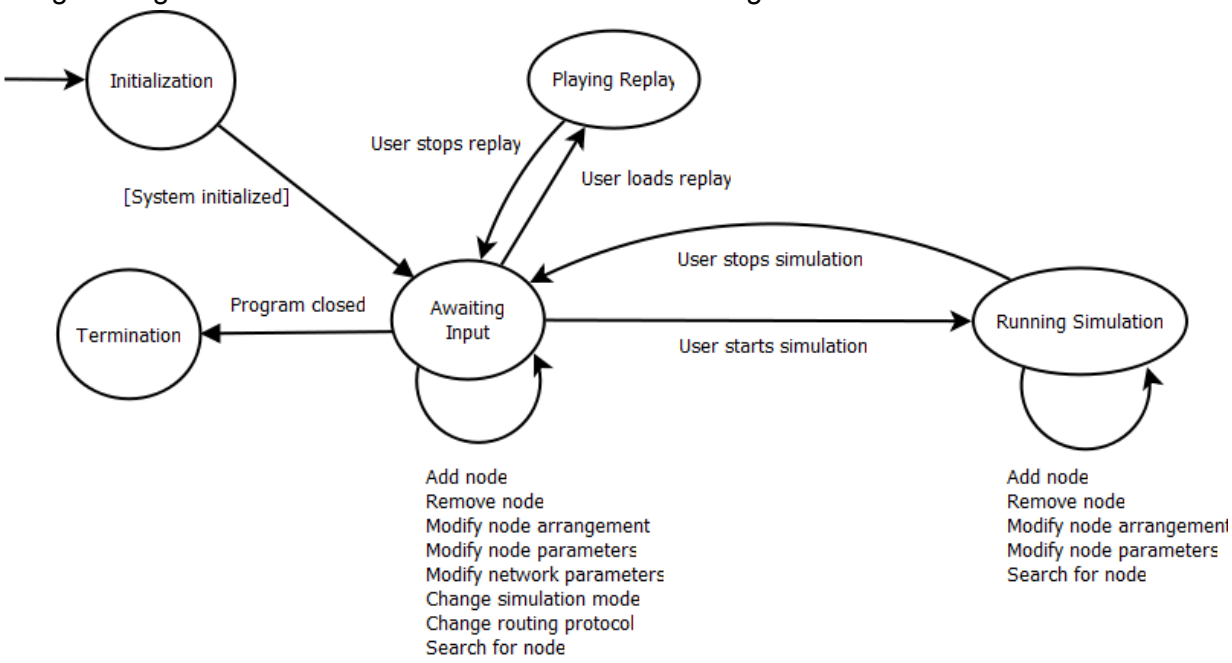
A wireless ad-hoc network simulator will need to be developed. The mission or objective of this project is to complete the software development life cycle and deploy a workable product that

can be used for educational purposes. The operational policies and constraints are to create a simulator that is platform independent and contains a balance between detail and performance.

The simulator Group 1 will develop is a modification of the current Ali and Masoud's AODV Simulator [6]. Ali and Masoud's simulator was developed in Java and Swing Group 1 will continue to use the same programming languages. Group 1 will build upon the current system to adhere to the provided requirements.

The simulator will contain multiple states, as shown below:

Image 1: High Level Wireless Ad-hoc Simulator State Diagram



The above state diagram [image 1] illustrates the at a high level the different types of states the simulator could encounter while in operation.

Major components of the simulator are the GUI front-end, which will be supported by Java Swing and the back-end logic, which will be developed in the Java programming language.

The GUI front-end will contain nodes that represent the wireless network receivers. Nodes can be created by using a drag and drop method. Once created, nodes can be moved within the display area and their properties can be adjusted by modifying the default node data within the node properties region. Nodes can also be deleted by clicking on the node and then selecting the delete selection from the node properties region. When simulating the data send process the simulator will clearly display when the process begins and ends. While sending data the simulation will display the data that is being processed in the data region. A network data region will be available to allow users to adjust the network settings of the simulation. A search will be available for users to easily find nodes. The GUI shall simulate an individual node

sending a message to another node or multiple nodes sending messages to other nodes.

The Java back-end will simulate the Ad-hoc On-demand Distance Vector (AODV) protocol [4] and the Destination Sequence Distance Vector (DSDV) protocol [5] during processing of messages. When creating, deleting or modifying nodes the Java logic will create, update or remove each node appropriately. While processing node messages and network data an accessible text file will retain a processing log. The each component's logic will be written in a modular format to allow for ease of maintenance and modularity.

Both components will be independent of each other, which allows developers with the option to use either component in their own simulation as they see fit.

The simulation will retain its current session's information until the application has been closed. Once closed the current session's data will be cleaned and removed the user's system. The only remaining data will be in the logging text file.

The simulation will run independently on the user's computer with no connections to the Internet. With no connections to the Internet the user's security and privacy are secured and not transferred across any actual networks.

4.0 Justification for and Nature of Changes

The objective is to provide users with a platform independent educational tool that can be reused with new modules and is simple enough for a computer science student to use and understand. The new system will need to be designed and built to exercise modularity in order to properly simulate the operation of ad-hoc networks under different protocols and display the ad-hoc network's operation using different GUIs. The new system also needs to be platform independent to allow students using different platforms to use the simulator.

The current system, Ali and Masoud's AODV Simulator, is not extensible due to the limitation of not being designed in a modular fashion. The AODV Simulator's graphical user interface contains deficiencies that will have to be revamped. The new mobile ad-hoc network simulator will simulate both the Ad hoc On-demand Distance Vector (AODV) and Destination Sequenced Distance Vector (DSDV) protocols and allow for additional protocols to be simulated.The GUI of the current system is unintuitive for users.The AODV Simulator's graphical user interface is convoluted and cryptic for users and the images that represent parts of the ad-hoc network are static making it difficult for the user to understand what protocol they are viewing. The new mobile ad-hoc network simulator will have an intuitive graphical user interface with tool tips to help explain what the user is viewing and animations that clearly depict the operation of protocols.

The modular design and implementation change is essential as this will allow the new system to use other modules for the GUI and simulate other protocols in the future. The changes to the GUI are secondary but necessary for the new system to be useful for students.

5.0 Concept for Modified System

Wireless ad-hoc network is a wireless network of computing nodes that is formed automatically without human intervention. Each node in the network has the ability to discover its neighbors and to construct routes to reach other nodes in the collection. It does not rely on any based station infrastructure for communication. Instead, hosts depend on each other to send and receive messages. In ad-hoc network, two hosts can exchange messages whenever they happen to be at communication range.

We selected to implement it in Java because Java is a standard, widely deployed framework. Consequently, the Java platform boasts a large number of optimized virtual machine implementations across many hardware and software configurations as well as a large number of compilers and languages that target this execution platform. The finished software product will require the ability to operate in several different environments. One of the requirements of the finished product is platform neutrality. By using the Java platform this will be easily accomplished because of Java's ability to run on most of the widely available operating systems in use. This will also negate the need to develop for things such as multiple browsers as the compiled product will run in the JRE on any of the common platforms (Windows, Linux, Mac OS).

The GUI and Module design will ensure the intuition and flexibility of the system.

With the application of DSDV protocols and additional protocols simulated, the system becomes more powerful.

6.0 Operational Scenarios

The product will provide predefined user input fields for things like coordinates and number of nodes. Things like radio buttons or toggle type tabs will prevent invalid user inputs.

We will use the principle of exception handling to verify user inputs against predetermined criteria/limits to avoid logical errors in calculations (i.e. entering a negative number of nodes, or exceeding a threshold of nodes which would cause the results to be unusable, or anything that would result in a divide by zero calculation).

The various states of operation are detailed in the state diagram (section 3, Image 1).

7.0 Summary of Impacts

As this is the first project the team will be working on the operational impacts that will affect the new system are not fully known, the main operational changes that could impact the team, users and the system:

- Cross integration of Modules with other Teams

The organizational impacts that could affect the team, users and the system include:

- The commitment of resources, in our case time

- Changes to the Team structure
- Skill level of the team
- Amount of communication

Impacts during system development that could affect the development process will not be fully known until the design process is completed but there are a few impacts that can be defined at this point:

- Changes to requirements set by the customer
- Availability of Resources
- Level of understanding of the requirements

8.0 Analysis of the Proposed System

Advantages of the new system:

- One of the goals of our new system is to improve on the ease of use of the simulator. Our new system will present a more intuitive user interface that is backed by useful documentation.
- Our system will feature a modular design. Programmers will have the ability to exchange modules with other teams for use in their own simulators.
- In addition to implementing the AODV protocol, we will include support for DSDV as well. Other protocols may be added in the future by extension of our program.

Disadvantages/limitations:

- The cost of designing a new system and the testing required to design a new, easy-to-use piece of software are minor disadvantages to creating a new system.
- Also, there could be features desired by end users that cannot be added simply because of the restricted timeline of our project.

Alternatives considered:

- We considered taking the source for AODV Simulator and modifying it to suit our needs, but this approach would involve learning how the code is organized, how it works, and how to properly modify and extend it. Since the project is not well-documented, we believe that there is a high risk that we will find the code to be difficult to decipher or modify. Therefore, this approach could end up hindering development.
- We also considered creating a web application using Java, but not everyone on the team has the necessary technical background, and time would be wasted on learning the basics of J2EE and web application development.
- We decided to design a new standalone application from the ground up using AODV Simulator simply as a reference. Designing the system from the ground up may require more time than adapting existing code, but it involves less risk as long as everyone is kept up to date on the requirements and the project architecture.

Glossary

AODV - Ad hoc on-demand distance vector is a reactive routing protocol that only requests a route when it needs one. Nodes are not required to maintain routes to destinations that are not communicating.

CONOPS - Concept of Operations document is a user-oriented document that describes system characteristics for a proposed system from the users' viewpoint.

DSDV - Destination-sequenced distance vector is a proactive hop-by-hop distance vector routing protocol where each network node maintains a routing table that contains the next hop to any reachable destination as well as the number of hops required to them. The routing table is updated via periodical broadcasts.

GUI - Graphical User Interface is an interface that allows a user to interact with a program through panels, buttons and other means not solely limited to text.

J2EE - Java Platform, Enterprise Edition or Java EE is a widely used platform for server programming in the Java programming language.

JRE - Java Runtime Environment is the execution platform for the JVM.

JVM - Java Virtual Machine is the platform which allows Java bytecode to be run (through the JRE) on multiple platforms

Mobile Ad Hoc Network - A network formed without any central administration which consists of mobile nodes that use a wireless interface to send packet data.

Node - a connection point on a network.