Application Dependent GEO-Routing Protocol on Multi Radio Networks

Smt. Geetha Jayaram Bhat [1]
[1] Associate Professor, DPGS-CEA,
NIE, Mysore, Karnataka,
India,
g.nettar@rediffmail.com

Mahanthesha U[2]
[2]Department of PG Studies, NIE,
Mysore, Karnataka,
India,
mahanthesh015@gmail.com

***Abstract*—** MANETS are provide with multiple radios which will be considered as the basis of future force communication networks envisioned for use in the war field network-centric applications. In this paper, we propose a new geographical routing protocol for such a different multi-radio, multi-band tactical MANET environment. This new protocol is designed to use many routing metric with different many radio interfaces, each of which runs on different frequency bands. This protocol gives high quality of service comparing to the existing GPRS technology. Simulation of the protocol results using the JProwler platform, java coding with swings, Netbeans IDE tool and windows OS.

Keywords- *Tactical ad hoc networks, Geographical routing*

*****

I. INTRODUCTION

We propose a new novel multi-metric geographical routing protocol with multi-radio for next generation TMMR. While location-based routing algorithms adopt a single radio, our scheme assumes that each node has multiple interfaces over multiple frequencies, resulting in different transmission ranges [1]. In this new protocol, a node is also assumed to share its own location information with neighboring nodes by exchanging periodic hello messages without control messages. [2].Problems arise when these existing algorithms are deployed in the multi-interface with different communication range. In this environment, the location routing protocols basically does not support multi-radio environment. Tactical Information Communication Network is one of the fundamental information communication systems which constitute the information mesh network of NCW [5].We proposes a novel multi-metric geographical routing protocol with different multi-radio TMMR. The networking structure in tactican information communication network is as shown in the Figure(1).

From the performance point of view, military networks are characterized by very strong requirements of data packet loss and delay compared to commercial networks, because it is more mission critical. Therefore, it is important to provide the appropriate routing path in accordance with tactical application requirement for Quality of Service (QoS). Also, since future tactical wireless networks should be able to coordinate services in dynamic network environments, how to make a selection of optimal interface depending on the type of demanding application is important[3].

We propose a novel multi-metric geographical routing protocol with multi-radio for next generation TMMR. While conventional location-based routing algorithms adopt a single radio, our scheme assumes that each node has multiple interfaces over multiple frequency bands, resulting in different transmission ranges. In our protocol, a node is also assumed to share its own location information with neighboring nodes by exchanging periodic hello messages without control messages. The main idea behind the proposed scheme is for node to select the most appropriate radio interface and hence routing metric corresponding to its on-demand data types[4].

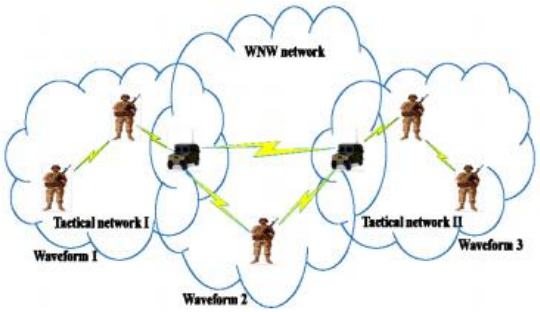

Figure (1) TMMR networking structure in tactican information communication network

II. HARDWARE AND SOFTWARE REQUIREMENTS

Netbean is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written primarily in Java and can be used to develop applications in Java and, by means of the various plug-ins, in other languages as well, including C, C++, COBOL, Python, Perl, PHP, and others. Netbean employs plug-ins in order to provide all of its functionality on top of (and including) the runtime system, in contrast to some other applications where functionality is typically hard coded.

The Netbean SDK includes the Netbean java development tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of metadata over a flat file space allowing external file

 ISSN: 2321-8169
1784 – 1789

modifications as long as the corresponding workspace "resource" is refreshed afterwards.

Hardware Requirements

Processors	:	Pentium IV
RAM	:	64 MB.
Storage	:	20GB.
Monitor	:	15”
Keyboard	:	Standard 102 keys

Software (Tools &Technologies) Requirements

Coding	:	Java
Platform	:	Jdk
Tool	:	Netbean ide
OS	:	Windows OS
Chart	:	JFreeChart
Simulator	:	Jprowler
Front end	:	Swings

Swing is a set of classes that provides more powerful and flexible components than are possible with the AWT. In addition to the familiar components, Swing supplies tabbed panes, scroll panes, trees, and tables. It provides a single API capable of supporting multiple look-and feels so that developers and end-users are not locked into a single platform’s look-and-feel. The Swing library makes heavy use of the MVC software design pattern, which conceptually decouples the data being viewed from the user interface controls through which it is viewed. Swing possesses several traits such as platform-independence, Extensibility, Component-oriented, Customizable, Configurable, Look and feel.

III. SYSTEM DESIGN

Fundamental Design Concepts

A set of fundamental design concepts has evolved over the past three decades. Although the degree of interest in each concept has varied over the years, each has stood the test of time. Each provides the software designer with a foundation from which more sophisticated design methods can be applied. The fundamental design concepts provide the necessary framework for “getting it right”. The fundamental design concepts such as abstraction, refinement, modularity, software architecture, control hierarchy, structural partitioning, data structure, software procedure and information hiding are applied in this project to getting it right as per the specification.

System Architecture

System architecture is the conceptual design that defines the structure and behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system. The System architecture is shown in Figure (2).

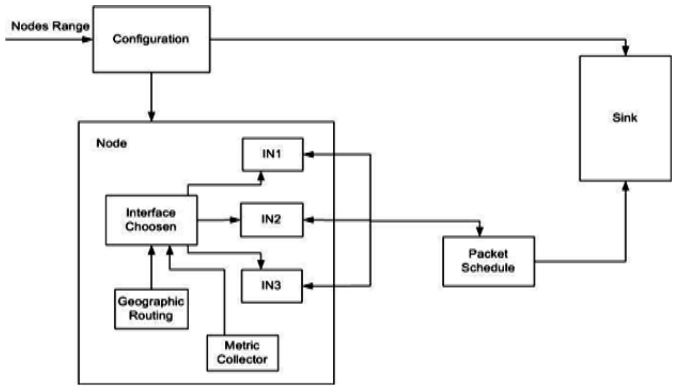

Figure (2). *The System Architecture*

Data Flow Diagram of the system

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

Level 0 Data flow diagram

A context-level or level 0 data flow diagram shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization. The Level 0 Data flow diagram is shown in Fig(3).

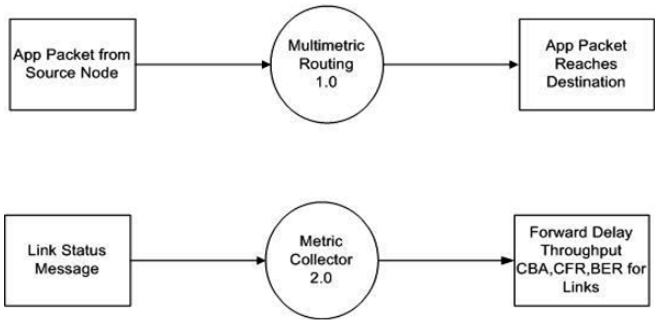

Figure (3) *Level 0 Data flow diagram.*

Level 1 Data flow diagram

The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of

the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system. The Level 0 Data flow diagram is shown in Fig(4).

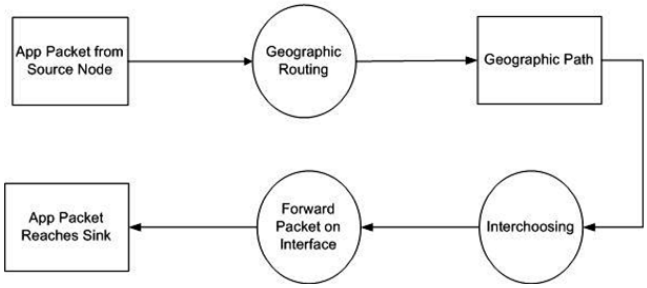

Figure (4). *Level 1 Data flow diagram.*

IV. IMPLEMENTATION

The implementation stage requires the following tasks.

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Evaluation of the changeover method.
- Correct decisions regarding selection of the platform
- Appropriate selection of the language for application development

.

Language used for implementation

Implementation phase should perfectly map the design document in a suitable programming language in order to achieve the necessary final and correct product. Often the product contains flaws and gets ruined due to incorrect programming language chosen for implementation.

In this project, for implementation purpose Java is chosen as the programming language. Few reasons for which Java is selected as a programming language can be outlined as follows:-

Platform Independence: Java compilers do not produce native object code for a particular platform but rather 'byte code' instructions for the Java Virtual Machine (JVM). Making Java code work on a particular platform is then simply a matter of writing a byte code interpreter to simulate a JVM. What this all means is that the same compiled byte code will run unmodified on any platform that supports Java.

Objects Orientation: Java is a pure object-oriented language. This means that everything in a Java program is an object and everything is descended from a root object class.

Rich Standard Library: One of Java's most attractive features is its standard library. The Java environment includes hundreds of classes and methods in six major functional areas:-

- Language Support classes for advanced language features such as strings, arrays, threads, and exception handling.
- Utility classes like a random number generator, date and time functions, and container classes.
- Input/output classes to read and write data of many types to and from a variety of sources.
- Networking classes to allow inter-computer communications over a local network or the Internet.
- Abstract Window Toolkit for creating platform-independent GUI applications.
- Applet is a class that lets you create Java programs that can be downloaded and run on a client browser.

Applet Interface: In addition to being able to create stand-alone applications, Java developers can create programs that can download from a web page and run on a client browser.

Familiar C++-like Syntax: One of the factors enabling the rapid adoption of Java is the similarity of the Java syntax to that of the popular C++ programming language.

Garbage Collection: Java does not require programmers to explicitly free dynamically allocated memory. This makes Java programs easier to write and less prone to memory errors.

Swing support: Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit. Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform.

Platform used for implementation

A platform is a crucial element in software development. A platform might be simply defined as "a place to launch software". In this project, for implementation purpose Windows XP platform is used & reasons for choosing this platform are Integrated Networking support, More stable and secure than previous version, Contain remote desktop connection and restore option, Enhanced device driver verifier, Dramatically reduced reboot scenarios, Improved code protection, Side-by-side DLL support, Windows File Protection, Preemptive multitasking architecture, Scalable memory and processor support, Encrypting File System (EFS) with multi-user support, IP Security (IPSec), Kerberos support, Smart card support, Internet Explorer Add-on Manager, Windows Firewall, Windows Security Center, Fresh visual design.

V. PERFORMANCE EVALUATION

ISSN: 2321-8169
1784 – 1789

Quality Assurance:

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confident that the product quality is meeting its goals. This is an "umbrella activity" that is applied throughout the engineering process. Software quality assurance encompasses:-

- Analysis, design, coding and testing methods and tools
- Formal technical reviews that are applied during each software engineering
- Mulitiered testing strategy
- Control of software documentation and the change made to it.
- A procedure to ensure compliance with software development standards.
- Measurement and reporting mechanisms.

The software quality assurance is comprised of a variety of tasks associated with seven major activities:-

- Application of technical methods.
- Conduct of formal technical reviews
- Software testing
- Enforcement of standards
- Control of change
- Measurement
- Record keeping and reporting

VI. INTERPRETATION OF RESULT

Following are the interpretation results getting in the simulation process.

End to End Delay :

It is the delay in transmitting voice traffic packets through wireless links plus the delay in the network interface queues due to network congestion. As shown in the figure(5) and figure(6), the End to End delay is less in multi metric while comparing to the GPRS technology. Hence the performace of the multimetric technology is very high comparing to the existing GPRS technology.

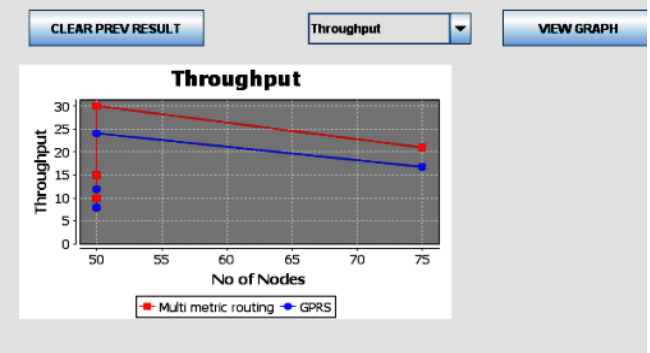

Figure (5): Illustration of the output End to End delay

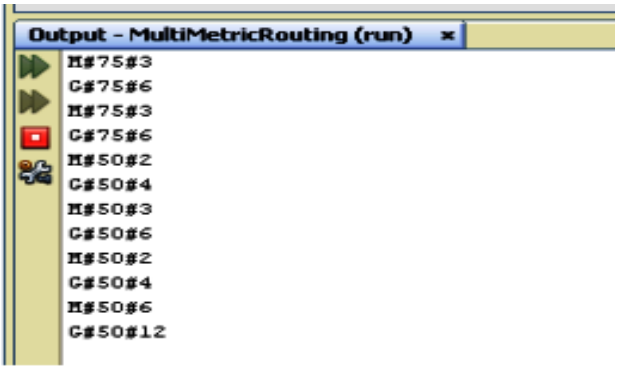

Figure (6): Output - multi metric routing of End to End delay

Packet Delivery Ratio :

It is the ratio of the number of packets received at the destination over the number of packets generated by the sources. As shown in the figure(7) and figure(8), the Ratio of the packet delivery is very high in multi metric while comparing to the GPRS technology. Hence the performace of this multimetric technology is very high comparing to the existing GPRS technology.

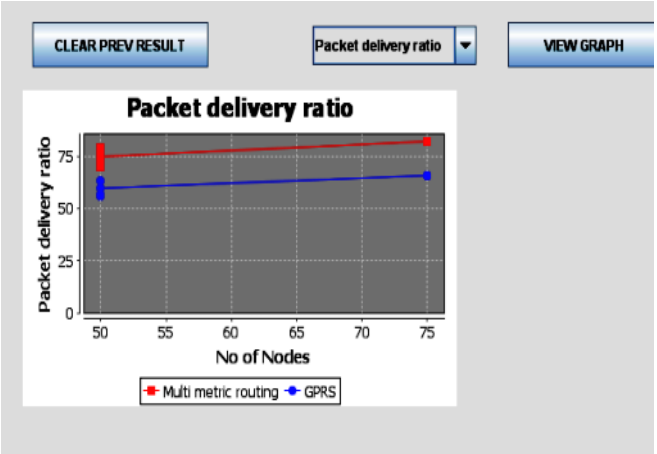

Figure (7): Illustration of the output Packet Delivery Ratio

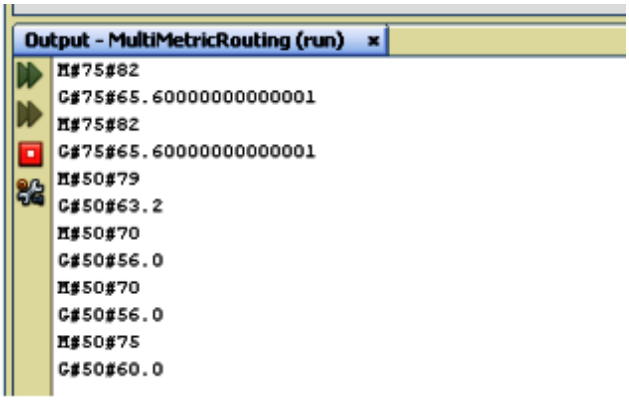

Figure (8): Output - multi metric routing of Packet Delivery Ratio

Throughput :

It is defined as the amount of successful received data measured in Megabyte per second, which reflects network reliability and performance for delivering video streaming. As shown in the figure(9) and figure(10), the throughput is very high in multi metric while comparing to the GPRS technology. Hence the performace of this multimetric technology is very high comparing to the existing GPRS technology.

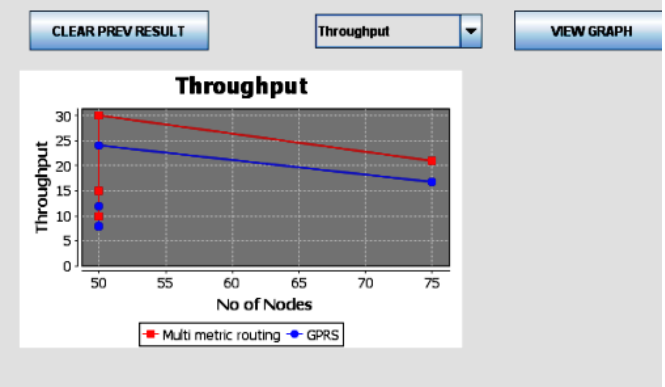

Figure (9): Illustration of the Throughput

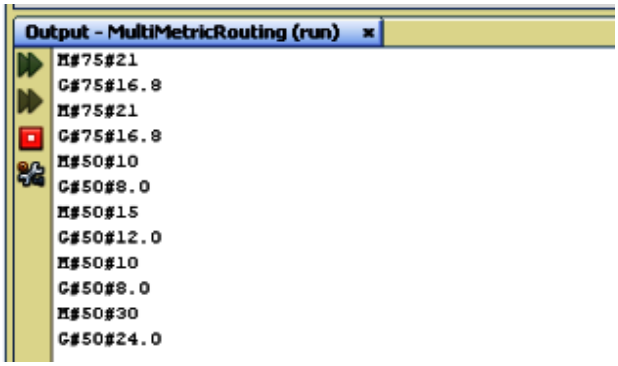

Figure (10): Output - multi metric routing of Throughput

VII. CONCLUSION

We have presented a new multi-metric geographical routing protocol for tactical ad hoc networks. In order to support simple but unique traffic characteristics of tactical operations, we proposed to utilize multiple routing metrics for three different types of data, i.e., for voice, video and short message data. After computing metric of interfaces, one of interfaces in the set can be chosen. Simulation preliminary results show that the proposed scheme outperforms than simple GPSR scheme in terms of end-to-end delay, packet delivery ratio and throughput with low overhead.

The End to End delay is less in multi metric while comparing to the GPRS technology. The Ratio of the packet delivery and the throughput are very high in multi metric while comparing to the GPRS technology Hence the performace of the multimetric technology is very high and useful when comparing to the existing GPRS technology.

As a future work, we plan to perform extensive simulation to explore the trade-offs made by different metrics in the protocol. Adapting the protocol for energy efficiency and mobility prediction are also our future inquiry.

REFERENCE

[1] J.L. Burbank, P.F. Chimento, B.K. Haberman, W.T. Kasch, "Key Challenges of Military Tactical Networking and the Elusive Promise of MANET Technology", IEEE Communications Magazine, vol. 44, Issue: 11 , 2006

[2] M. S. Hasan, M. LaMacchia, L. Muzzelo, R. Gunsaulis, "Designing the joint tactical radio system (JTRS) Handheld, Manpack, and small Form Fit (HMS) Radios for interoperable networking and waveform applications," in Proc. of IEEE MILCOM, 2007

[3] JTRS HMS Small Form Fit Radios, 2010 General Dynamics http://www.gdc4s.com/documents/D-JTRSSFF-6-0210w.pdf.

[4] J. Redi, R. Ramanathan, "The DARPA WNaN network architecture," in Proc. of IEEE MILCOM, 2011

[5] D. Kim, "Load Balancing with Mobile Base Stations in Tactical Information Communication Networks," in Proc. of IEEE WCNC, 2011

[6] M.E.M. Campista, P.M. Esposito, I.M. Moraes, L.H.M. Costa, O.C.M. Duarte, D.G. Passos, C.V.N. de Albuquerque, D.C.M. Saade, M.G. Rubinstein, "Routing metrics and protocols for wireless mesh networks," IEEE Network, vol. 22, 2008

[7] E. Charles, B. Perkins, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in Proc. Of SIGCOMM, 1994

[8] C.E. Perkins, E.M. Royer, "Ad-hoc On-Demand Distance Vector Routing," in Proc. of IEEE Workshop On Mobile Computing Systems And Applications, pp. 90--100, 1997

[9] B. J. David, A. M. David, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing book Kluwer Academic Publishers, pp. 153--181, 1996

[10] Y.B. Ko, N. H. Vaidya, " Location-Aided Routing (LAR) in mobile ad hoc networks," Wireless Networks, vol. 6, no. 4, pp. 307—321, 2000

[11] B. Karp, H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in Proc. of MobiCom, 2000

[12] L. Blazevic, L. J. Boudec, S. Giordano, "A Location-Based Routing Method for Mobile Ad Hoc Networks," IEEE Transactions on Mobile computing, vol. 3, no. 4, pp. 97--110, 2005

[13] Liao,W. H., Sheu, J.P., Tseng, Y.C., "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," Telecommunication Systems, vol.18, no.1–3, pp. 37--60, 2001

[14] A. A. Sakr, A. M. Kozae, M. Elshafae, M. R. Ibraheem, "Optimizing Network Routing by Deducing a QoS Metric Using

Rough Sets," International Journal of Engineering Science and Technology, vol. 2(7), pp. 2101--3110, 2010

[15] D. Chen, P.K. Varshney, "A survey of void handling techniques for eographic routing in wireless networks," IEEE Communication Survey & Tutorials, vol. 9, no. 1, 2007.

[16] B. N Cheng, J. Wheeler, L. Veytser, "Radio-to-Router interface technology and its applicabliity on the tactical edge",IEEE communication Magazine, Vo. 50 , pp. 70--77 , 2012

[17] Office of the Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer, "Net-Centric Checklist," August 14, 2004 Ver. 2.2.

[18] H. Badis, A. Munaretto, K. A. Agha, G. Pujolle, "QoS for Ad hoc Networking Based on Multiple Metrics: Bandwidth and Delay. in Proc. Of MWCN, 2003

[19] L. Cao, K. Sharif, Y. Wang, T. Dahlberg, "Adaptive Multiple Metrics Routing Protocols for Heterogeneous Multi-Hop Wireless Networks," in Proc. of CCNC, 2008

[20] R. Kumar, M. Misra, A. K. Sarje, "A Routing Protocol for Delay-Sensitive Applications in Mobile Ad Hoc Networks," in Proc. Of ISAUHC, 2006