Army Research Laboratory

The ARL Topodef Tool—Concept and Usage

by Binh Nguyen

ARL-TR-5002 October 2009

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-5002 **October 2009**

The ARL Topodef Tool—Concept and Usage

Binh Nguyen
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*
October 2009

2. REPORT TYPE
Final

3. DATES COVERED (From - To)

4. TITLE AND SUBTITLE
The ARL Topodef Tool—Concept and Usage

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Binh Nguyen

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: RDRL-CIN-T
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-5002

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report describes the functional features of the U.S. Army Research Laboratory (ARL) Topodef tool, a graphical tool and method for custom-designing network topology of a mobile ad hoc network to support emulation. The tool is also known as a scenario-generator capable of generating a set of mobility traces or a set of textual definitions of a dynamic network topology. The tool is being used to edit a set of existing mobility traces to meet a research requirement. Instructions for using the tool are documented in this report. The tool is now being used at government and industrial laboratories. Known users of the tool include the U.S. Naval Research Laboratory (NRL), Telcordia Inc., and Sparta Inc.

15. SUBJECT TERMS
Topodef tool, topology, scenario-generator

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	36	Brian Rivera
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER (*Include area code*) (301) 394-2298

Contents

List of Figures

INTENTIONALLY LEFT BLANK.

1. Introduction

1.1 Background

A mobile ad-hoc network (MANET) is an autonomous system of mobile nodes capable of communicating with one another wirelessly. When a node moves within the radio-transmission range of another node, a communication link is established, and when it moves farther away, its existing communication link is destroyed. A network topology is a snapshot in time showing the positions of the participating nodes and their communication links, which vary from time to time and may consist of both bidirectional and unidirectional links. A more complete description of the characteristics of MANET can be found in (*1*).

Conducting a MANET experiment can be done in at least three different ways: actual field exercises and measurements, simulation, and emulation. An actual field exercise is the most expensive way in terms of time, effort, and resources. Therefore, computer simulation programs and emulation test beds are preferred methods for predicting the performance of a MANET system under test. Simulation provides a controllable and repeatable virtual environment in which theoretical concepts are implemented and evaluated. Emulation is between the two; it provides a controllable physical environment in which experimentation can be conducted.

Whether simulation or emulation is used, having a specific static and dynamic network topology meeting an exact need is desirable in all phases of research, development, testing, and evaluation of MANET technologies and their applications. During early phases, having a controllable topology is highly desirable because a new technique, algorithm, or mechanism that is being developed may not be ready to operate in any topology. During the final phases, it can be used to test and evaluate a newly developed technology in order to corroborate its performance claims.

A specific network topology meets at least the following requirements of a research need:

- The exact number of participating nodes
- Their relative geographical positions
- The number of communication links connected to each node

Specific mobility-pattern data are time-stamped mobility traces that meet an exact need. Accordingly, satisfying these requirements necessitates the development of a way for designing, editing, manipulating, verifying, and animating desired network topologies, and creating desired mobility-pattern data of a virtual MANET system.

The U.S. Army Research Laboratory (ARL) Topodef tool is developed to meet these special requirements and mainly to support emulation, although its outputs can also be used to feed a simulation tool. The main outputs of the tool are a topology-definition file (TDF) and a set of *log* files. TDF files are used to drive the TEALab test bed (*2*), and *log* files, the MANE test bed (*3*). The two test beds are functionally equivalent, but they differ from one another architecturally and thus require different types of inputs to function.

The TEALab test bed is a distributed system consisting of networked computers acting as MANET hosts and running the Linux operating system (OS) interconnected via a common networking environment—e.g., wired or wireless Ethernet. Driving TEALab requires a TDF file containing an ordered list of textual specifications of a network topology at a certain time. The contents of the TDF file are then interpreted and translated by a host-based topology-scenario manager (TSM) into *iptables* commands (packet-filtering rules) to control the in-flows of network packets arriving at each node, effectively managing network connectivity and creating a virtual network topology. Thus, each node in the TEALab test bed must have a TDF file, and the execution of the TSM is synchronized with all the participating nodes.

On the other hand, the MANE test bed is a centralized system connected to a wired Gigabit Ethernet network and consisting of one or more servers. Each server provides a central connection point for a set of computers called "test nodes" (TNs), and thus it controls the flow of network traffic generated by the connected test nodes. MANE servers determine the existence of a communication link at run-time using a range model and the distance between two nodes. Thus, MANE does not need any specifications of a link because it figures them out by using geodetic positions of each node at a certain time. Each TN must have a *log* file, which stores its own mobility traces (time-stamped positions).

1.2 Scope

This report documents the ARL Topodef tool by describing its background, features, benefits, and instructions for using the tool. The intended audience of this report includes mainly the users of the tool. The main purposes of this document are as follows:

- Reporting the development of the ARL Topodef tool
- Recommending possible enhancements to the tool

The next section provides an overview of the ARL Topodef tool and explains how it can be used to create mobility scenario files. The last section summarizes special features of the tool and presents a list of possible enhancements. The appendix provides a list of terminologies that are used in this document.

2. The ARL Topodef Tool

2.1 Description

The ARL Topodef tool is graphical tool and method for custom-defining a network topology. Its current implementation uses the Python programming language to express logical and mathematical operations and the *Tkinter* modules/libraries that come with the language to create graphical user interfaces (GUIs). The ARL Topodef tool logic, thus, relies heavily on the functional features of the GUI, especially the exploitation of user-created *events*.

An event is generated whenever the left button or the right button of a computer mouse is pressed. The left button is used to select a graphical object being displayed on the screen or to drag an object on the screen. Dragging is performed while the left button is pressed and the cursor is moved to another place on the screen. If there is no event, then ARL Topodef tool enters a waiting state, doing nothing but waiting for an event to occur.

The ARL Topodef tool reacts to each user-generated event by binding a special function to a particular event. The name of the special function is registered with the GUI system, telling it to use the registered function to handle a particular event. The special function is often called a *callback* function or an event handler. For example, the onLeftClickReleased() function is registered with the system to handle the event associated with the releasing of the left mouse-button. Whenever an event occurs, the employed GUI system invokes the callback function and passes an *event* object whose attributes include the location on the canvas where the event occurred—e.g., *event.x, event.y*.

The ARL Topodef tool has two separate windows, as depicted in figures 1 and 2. The smaller window is populated by graphical objects—e.g., icons and action buttons. The larger one is used primarily to display a network topology with or without a background image complying with the graphics interchange format (GIF). Figure 1 shows the appearance of the tool on a computer running the Microsoft Windows XP OS, and figure 2 shows its appearance on a computer running the Red Hat Linux OS.

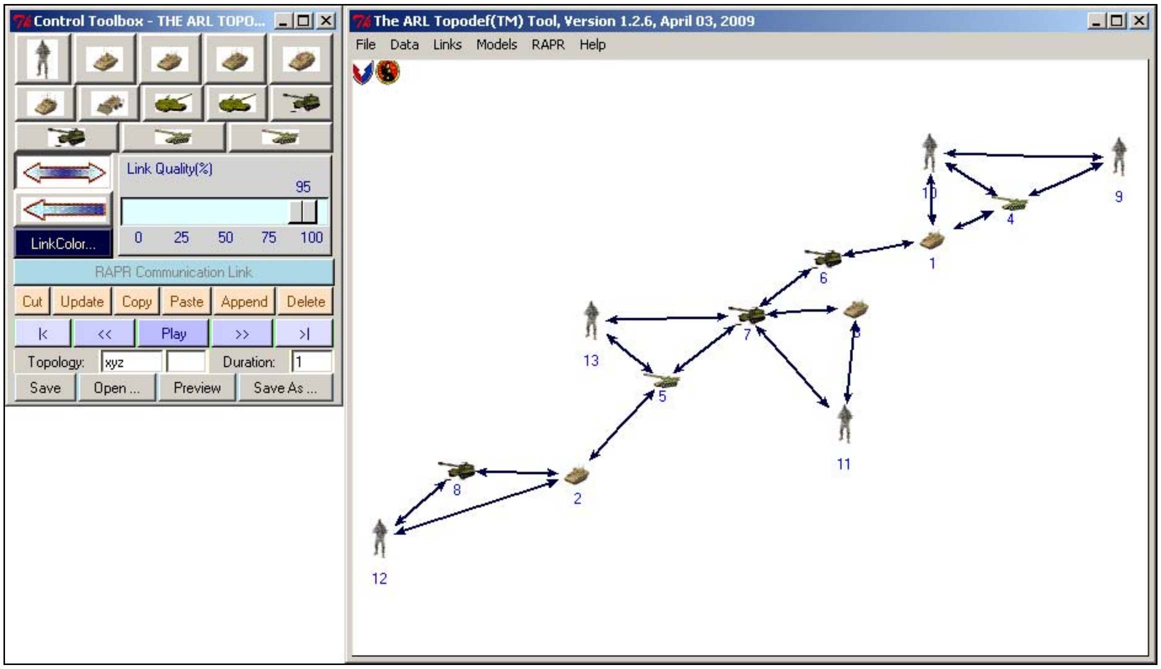


Figure 1. The ARL Topodef tool running on Microsoft Windows XP environments.

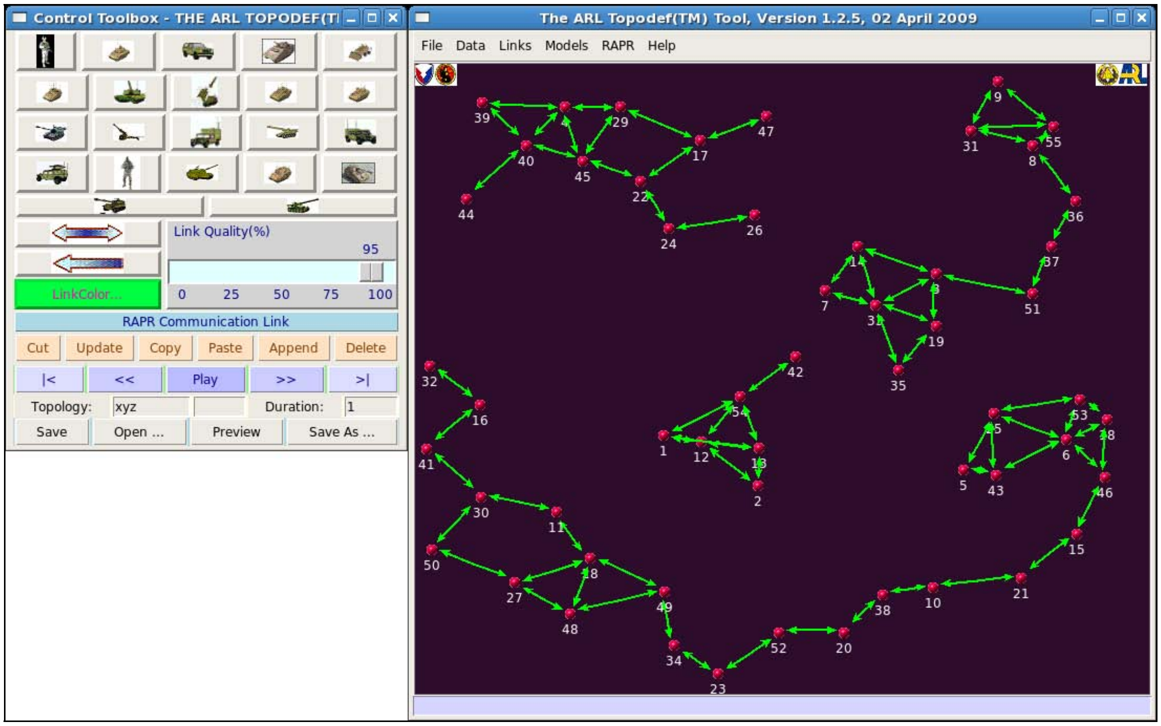
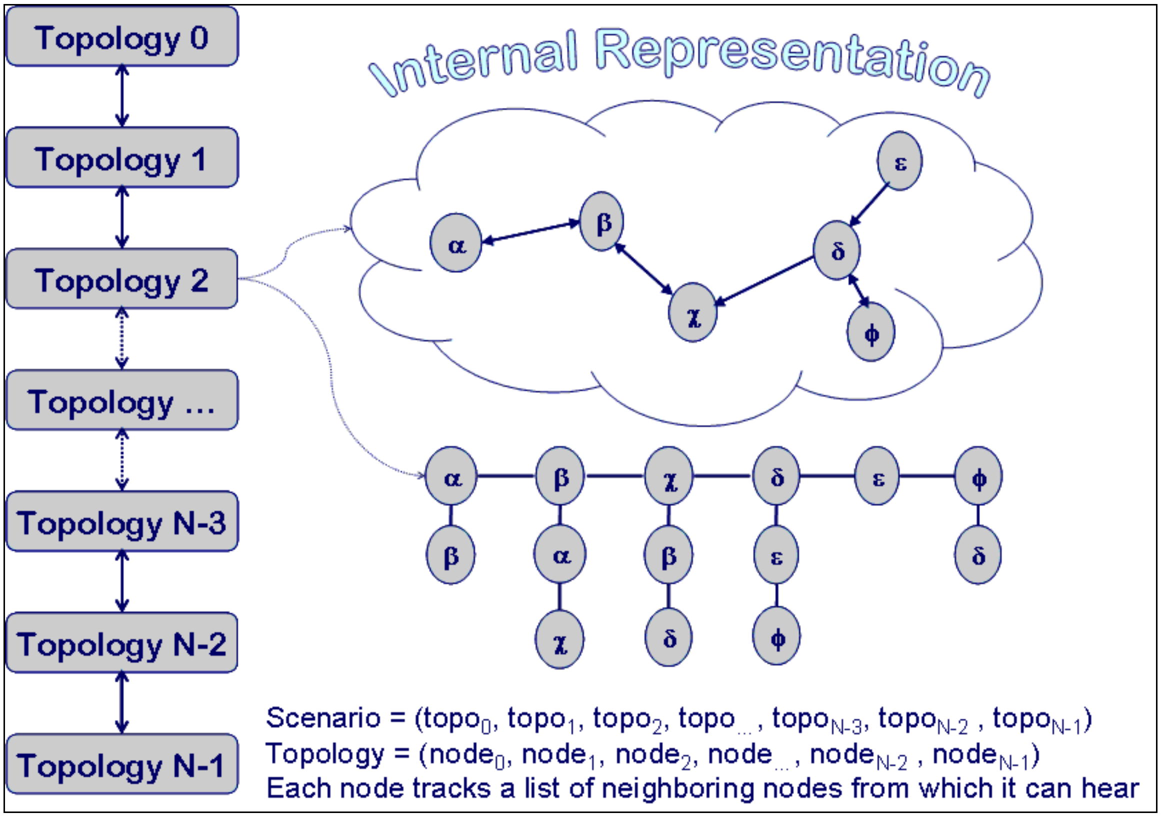


Figure 2. The ARL Topodef tool running on Red Hat Linux environments.

2.2 Implementation

The tool stores the specifications of each discrete network topology in a Python mutable composite data structure called *list*, which is an ordered collection of topology specifications, as shown in figure 3. Within the context of this particular implementation, the *list* can be considered metaphorically as an electronic flipbook in which every page has the required specifications of a network topology. Each network topology is associated with an ordinal number called *index* indicating its ordered position in the *list*, which is used to select a specific topology in the *list*. Once a topology is stored in the *list*, it can be retrieved, modified, updated, removed from the *list*, or relocated to a different location in the *list*. An n-topology scenario can then be permuted to create n-factorial (n!) different scenarios. An animation of the dynamically changing topologies is accomplished by successively displaying each topology in the *list*, automatically or manually, using the navigation tools described in the next section.

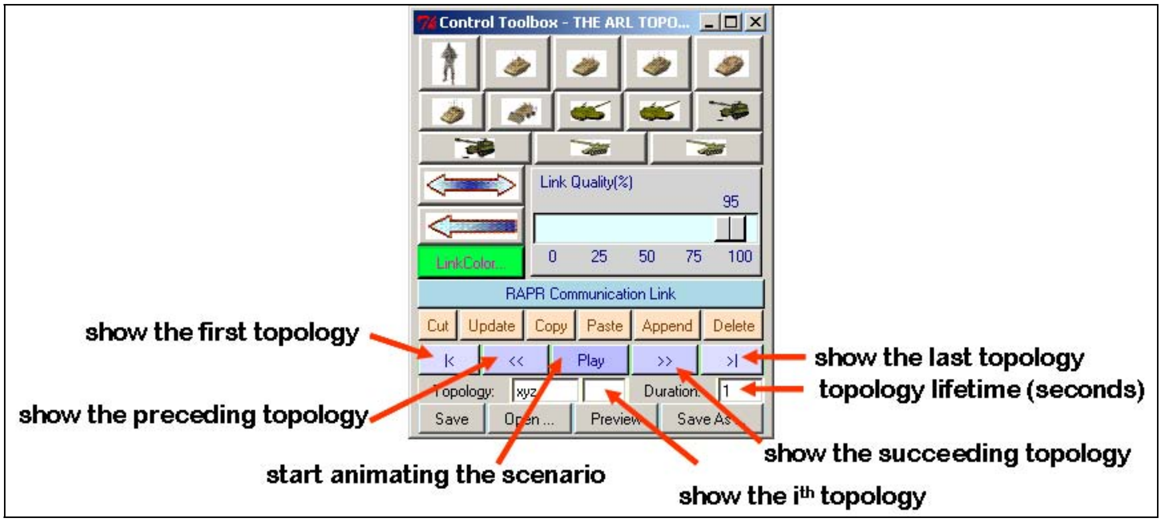


Scenario = ($topo_0$, $topo_1$, $topo_2$, $topo_{...}$, $topo_{N-3}$, $topo_{N-2}$, $topo_{N-1}$)
Topology = ($node_0$, $node_1$, $node_2$, $node_{...}$, $node_{N-2}$, $node_{N-1}$)
Each node tracks a list of neighboring nodes from which it can hear

Figure 3. Internal representation of a dynamic network topology.

2.3 Navigation Tools

The ARL Topodef tool provides a set of graphical objects for traversing the internal *list* of discrete network topologies, as shown in figure 4. The stored data can be viewed consecutively or randomly. The navigation tool has five action buttons and one entry box.



show the first topology

show the preceding topology

start animating the scenario

show the last topology

topology lifetime (seconds)

show the succeeding topology

show the i^{th} topology

Figure 4. Navigation tool for traversing a list of topologies.

The Entry Box. The entry box is used for displaying a specific network topology being stored in the *list*. The value of an entry is an integer representing the ordinal number of the associated topology—e.g., topologies[i] or scenario[i]. The entry box is also used to indicate the ordinal number of the topology being displayed on the screen.

The |< Button. The first topology in the scenario is displayed when this button is pressed, i.e., current_topology = scenario[0].

The << Button. The preceding topology is displayed when this button is pressed, i.e., current_topology = scenario[i-1], if the index i is greater than 0; otherwise, current_topology = scenario[0].

The >> Button. The succeeding topology is displayed when this button is pressed, i.e., current_topology = scenario[i+1], if the index *i* is less than the length, i.e., i < length(list) *-1* ; otherwise, *current_topology = scenario[-1]* (the last item in the list).

The >| Button. The last topology is displayed when this button is pressed, i.e., current_topology = scenario[-1].

*The Play Button***.** This button is used to control an animation; it toggles between the "Play" and the "Stop" labels every time it is pressed. A dynamic network topology is graphically animated if this button is pressed when it is labeled Play**.** At the start of an animation, the label is changed from Play to Stop, indicating that the animation can be stopped. When it is stopped or at the end of an animation, the label is changed to Play.

3. Operational Modes

The ARL Topodef tool can be used in one of the three design modes: manual, automatic, and semi-automatic modes. The manual mode provides the flexibility to create and position a node and to choose a node icon. The automatic mode uses a built-in mobility model and a range model to create a scenario. The semi-automatic mode offers the user a way to create a scenario by creating a path for a node to travel and by mixing the previous two operational modes. Regardless of the selected mode, the end result can be edited until a desired network topology is satisfactorily defined. Figure 5 shows various methods for designing a new scenario or for editing an existing one. The subsections that follow describe more details about each operational mode and how it can be used.

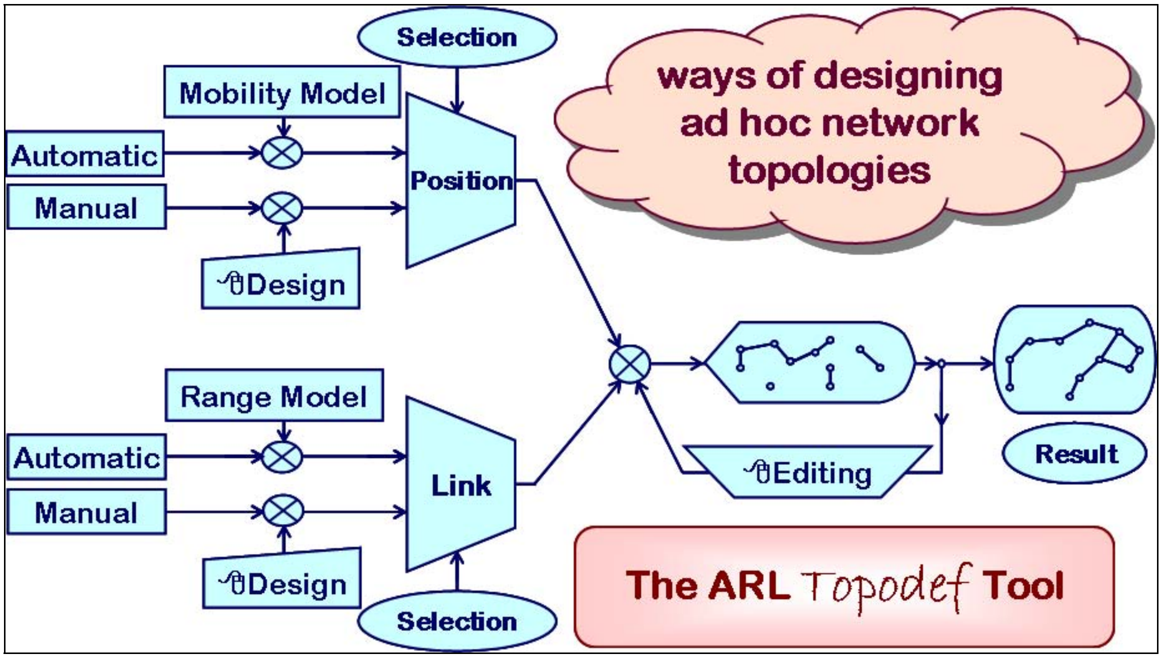


Figure 5. Various ways to create a specific network topology.

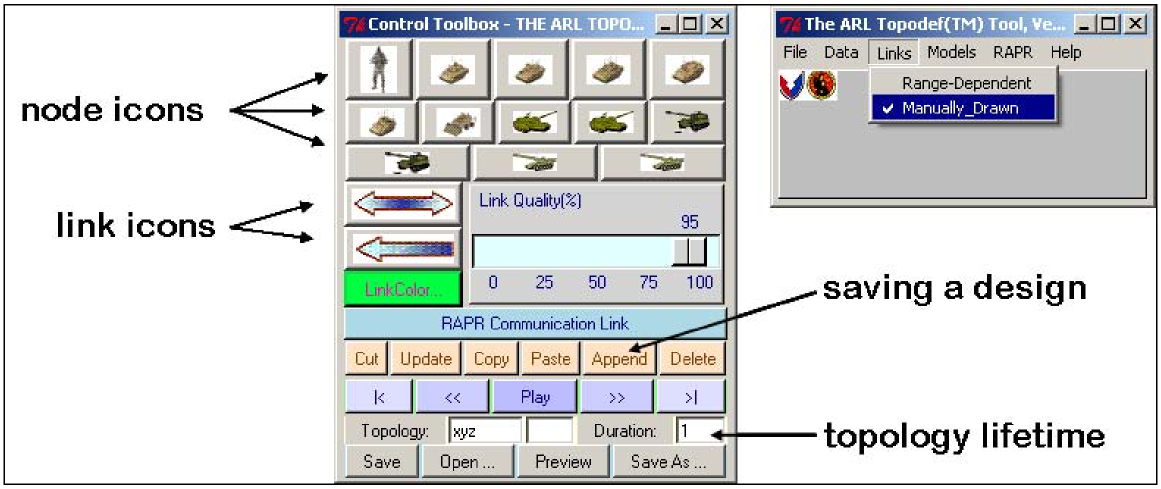
3.1 Manual Mode

The manual mode is designed to create an arbitrary number of nodes, their locations, and their interconnection links. This operational mode is intended for supporting the emulation of a MANET in a test bed that uses a packet-filtering system tool to control the incoming data packet at a given network interface of a test node—e.g., the TEALab test bed. This type of test bed can create a data link between any two nodes, regardless of the distance between them.

The positions of the participating nodes are used mainly for visualization purposes, and they can be extracted from the defined topologies for use in other types of test beds; however, the interconnection links between any two nodes may be different, depending on whether the test bed would consider the node positions and their relative distances. For example, the distance between any two nodes is a variable in the MANE test bed; therefore, although the node positions can be used in the MANE test bed, their interconnection links may be different.

The manual mode offers a way for the manual creation of nodes and their interconnection links. The process involves the setting and selection of the following items:

- Setting the tool to operate in the manual mode by selecting the option to manually create interconnection links (figure 6)
- Selecting a node icon or a link icon; e.g., a Soldier, a tank, a single-headed or double-headed arrow (figure 6)
- Creating and positioning the mobile nodes on the screen and establishing a link between two nodes



✓ Manually_Drawn

Link Quality(%) 95

Topology: xyz

Duration: 1

Figure 6. Manual creation of nodes and links.

3.1.1 Node Creation

The creation of a node is accomplished by simply moving the mouse cursor to the screen/canvas and clicking the left mouse button at the desired cursor location. When the left mouse button is released, the selected node icon is placed on the screen at the current cursor location. The process is repeated until the number of nodes satisfies a design requirement.

3.1.2 Link Creation

The creation of a link consists of performing two steps: selecting an icon representing a communication link and manually creating a link between two nodes. The toolbox has two link icons representing unidirectional link (⟵) or a bidirectional link (⟷). Creating a link is simply done by dragging the mouse cursor between two nodes. The head of the arrow rests on the outer edge of the icon representing the receiving node, and its tail rests on that of the transmitting node, as shown in figure 7. The process is repeated until a set of requirements for a specific topology is met—for example, the number of links connected to a particular node.

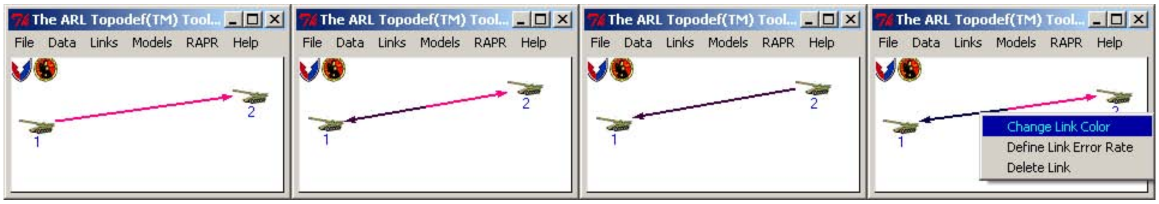


Figure 7. Unidirectional and bidirectional links between two nodes.

3.1.3 Topology Creation

The creation of a network topology is accomplished by creating the number of nodes and the number of links as described in the above sections. The process is completed by assigning a numerical value indicating the lifetime (duration) of the topology and by pressing the Append button to store the design internally. The default lifetime is one second, the lowest value, and it can be re-assigned by typing an integer into the entry box, as shown in figure 6.

3.1.4 Scenario Creation

The manual creation of a scenario is accomplished by repeatedly creating discrete network topologies, as described previously. Whenever the Append button is pressed, the currently displayed topology is inserted into the end of the *list*.

3.2 Automatic Mode

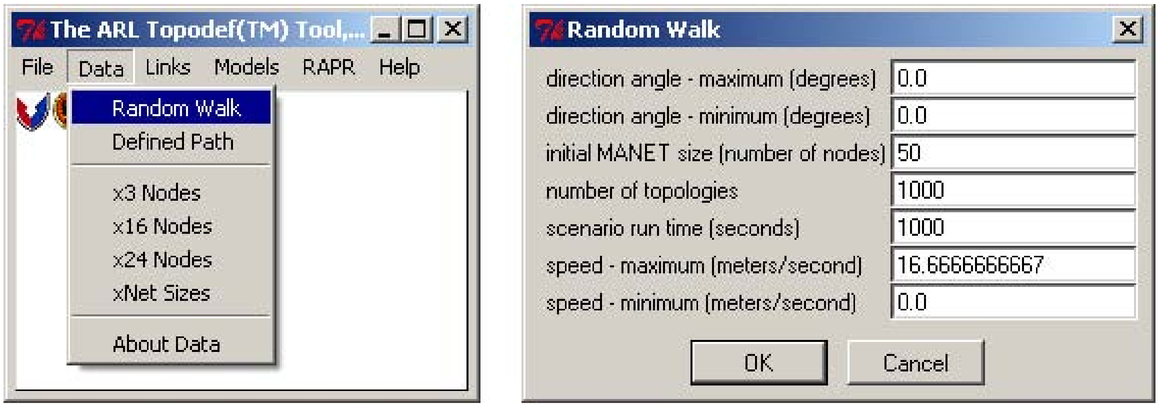
The automatic mode is designed to create a dynamic MANET scenario using a built-in mobility model and, optionally, a range model. The tool now provides a widely used mobility model called “random walk” to generate node movement positions and two range models—the free-space loss and the fixed-range model—to determine the existence of a link between any two nodes. The following subsections further describe the models and their uses in the ARL Topodef tool.

3.2.1 Node Movement

A mobile scenario can be generated using the random-walk model, which uses the Python pseudo-uniform random number generator to set the direction and the speed of a mobile node between their specified minimum and maximum values. The speed of a mobile node remains constant throughout the generation of a scenario, but the direction is reversed if the node reaches a border of the screen.

Whenever the random-walk model is selected, it displays a window showing the parameter name-value pairs, which can be changed to produce different outputs, as shown in figure 8. The model performs the following actions after the OK button is pressed:

- Calculating the lifetime (duration) of each network topology by dividing the scenario run time by the number of topologies, which should evenly divide the scenario run time. The number of topologies also indicates the number of moving steps.
- Randomly selecting a node icon to represent a newly created node if the total number of nodes is fewer than 48 or a small icon (e.g., a ball), otherwise.
- Generating a virtual MANET by creating the number of participating nodes specified by the user—e.g., 50 mobile nodes, as shown in figure 8.
- Generating and displaying node movement on the screen.



direction angle - maximum (degrees)	0.0
direction angle - minimum (degrees)	0.0
initial MANET size (number of nodes)	50
number of topologies	1000
scenario run time (seconds)	1000
speed - maximum (meters/second)	16.6666666667
speed - minimum (meters/second)	0.0

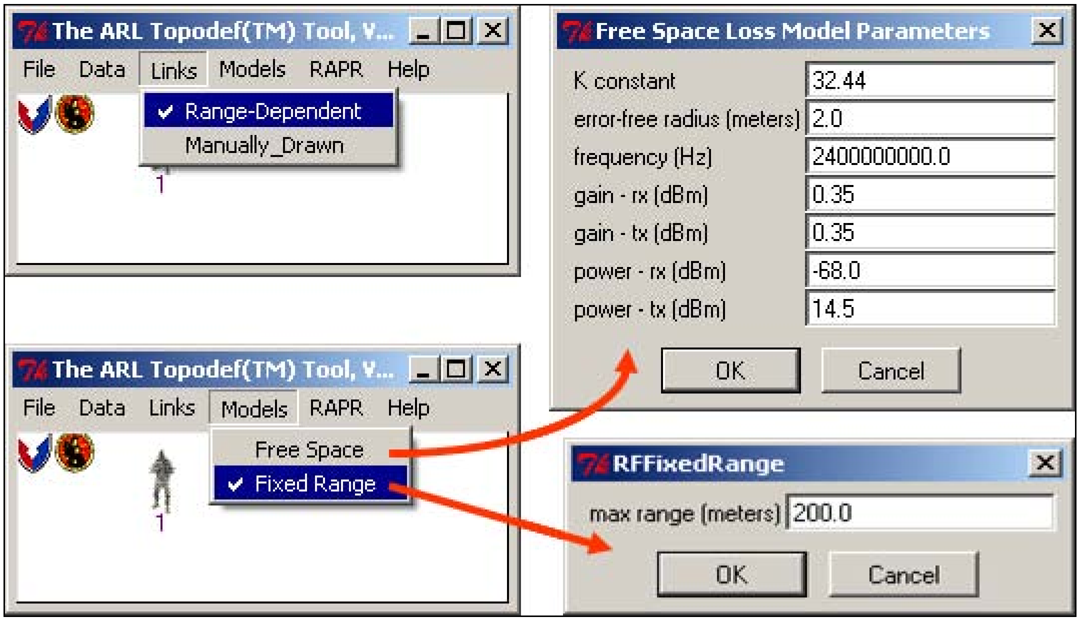
Figure 8. The random-walk model.

3.2.2 Link Creation

Creating and visualizing dynamic communication links is an optional feature in the generation of a scenario. The feature is enabled when the menu item “Range-Dependent” in the Links pull-down menu is selected, as shown in figure 9. The automatic link creation relies on a range model, which is the fixed-range or the free-space-loss model.

The fixed-range model uses an arbitrary value that indicates the maximum radio-transmission range; whereas, the free-space-loss model calculates this value using radio parameter values. To determine whether a link should exist between any two nodes, both models compute the distance between two nodes and then compare it with the maximum transmission range. If the two nodes are within range, then a link is established; otherwise, it is destroyed if it already exists.

This feature is computationally expensive because the tool must compute the distance and the radio transmission range for every pair of nodes. Doubling the number of nodes quadruples the number of times the tool must compute.



K constant 32.44
error-free radius (meters) 2.0
frequency (Hz) 2400000000.0
gain - rx (dBm) 0.35
gain - tx (dBm) 0.35
power - rx (dBm) -68.0
power - tx (dBm) 14.5

max range (meters) 200.0

Figure 9. Automatic creation of links.

3.2.3 Topology/Scenario Creation

A series of discrete network topologies is automatically created to form a dynamic scenario. At the beginning, when the first topology is formed, all participating nodes are randomly created and positioned on the screen, and all possible interconnection links are computed and visualized. Subsequent topologies are formed by newly updated node positions and computed interconnection links. At the end of each topology creation, the descriptions of the currently displayed topology are automatically stored in the internal *list* of the tool. In summary, to create and visualize a dynamic topology of a MANET, the “Range-Dependent” menu item must be selected, and the parameter values of the selected models should be verified or updated.

3.3 Semi-Automatic Mode

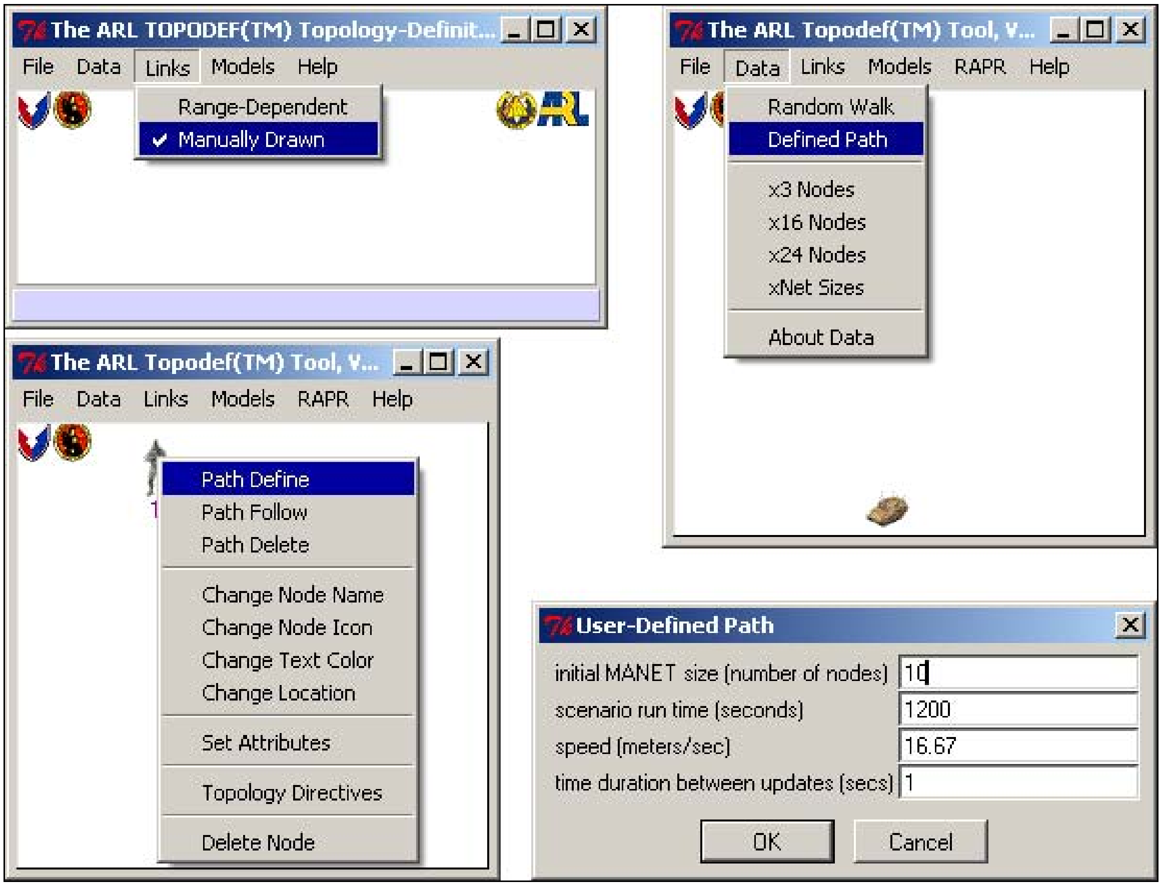
The semi-automatic mode is designed to create a specific network topology by combining the manual or automatic modes described in the preceding sections. This section presents a way for creating a path for a node to travel and two other possible ways to create a specific network topology of a MANET under test.

3.3.1 Node Traveling Path

Defining a path for a node to travel involves the following steps, requiring the use of the graphical objects depicted in figures 10 and 11:

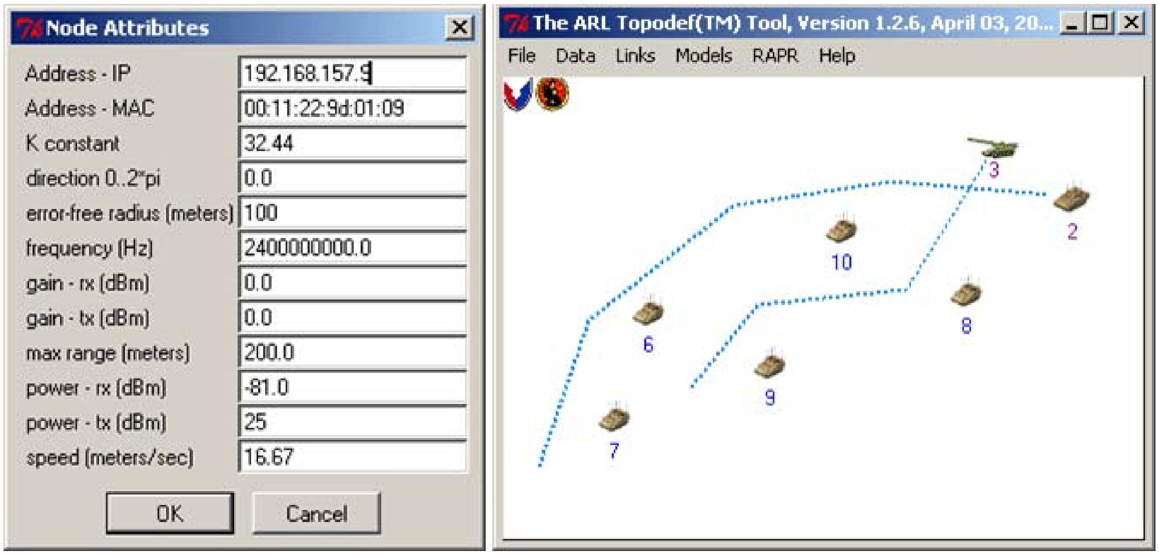
- Selecting the option to let the links be arbitrarily established (manually drawn) or dependent on the range of their radios (range-dependent). If the latter option is selected, then the distances between the moving node and its neighbors are calculated and used to determine the possibility for a link to exist.

- Selecting the option to let a mobile node travel on a defined path, and when a window is popped up, specifying the number of nodes, scenario run time (T), initial traveling speed for all the participating nodes, and time between updates (t). (The number of topologies (n) is obtained by dividing T with t (n = floor(T/t).)
- Changing the traveling speed of a particular node, if necessary, by right-clicking the mouse button on the node icon and selecting the Set Attributes menu item, and entering the new speed in the pop-up window that shows a list of node attributes that can be changed.
- Defining a path for a node to travel by first right-clicking the mouse button on the node icon and selecting the Path Define menu item, then positioning the cursor to a location on the screen and pressing the left mouse button. The traveling path can be extended by continuing to move the cursor to another position and pressing the left mouse button.
- Letting the mobile node to travel on its freshly defined path by right-clicking the mouse button on the node icon and selecting the Path Follow menu item. The mobile node then moves a *maximum* of n steps. If it moves fewer than n steps, its path is short and can be lengthened if desired by repeating the preceding procedure to define a path.
- Changing the appearance of the travel path is also possible by pressing the right mouse button on the travel path to pop up a menu of items from which one can be chosen—e.g., changing its color or line width.



initial MANET size (number of nodes)	10
scenario run time (seconds)	1200
speed (meters/sec)	16.67
time duration between updates (secs)	1

Figure 10. User-defined travel paths.


Node Attributes
OK
Cancel
The ARL Topodef(TM) Tool, Version 1.2.6, April 03, 20...
File Data Links Models RAPR Help
3
2
10
6
8
9
7

Address - IP	192.168.157.9
Address - MAC	00:11:22:9d:01:09
K constant	32.44
direction 0..2*pi	0.0
error-free radius (meters)	100
frequency (Hz)	2400000000.0
gain - rx (dBm)	0.0
gain - tx (dBm)	0.0
max range (meters)	200.0
power - rx (dBm)	-81.0
power - tx (dBm)	25
speed (meters/sec)	16.67

Figure 11. Editable node attributes and traveling paths.

3.3.2 Manual Node Creation, Automatic Link Creation

This method combines the procedure described in sections 3.1.1 and 3.2.2, which respectively describe the procedure for the manual creation of a node and for the automatic creation of a link between two nodes.

3.3.3 Automatic Node Creation, Manual Link Creation

This method combines the procedure described in sections 3.2.1 and 3.1.2, which respectively describe the procedure for creating a node automatically and for letting the links be created manually (and arbitrarily).

3.4 Topology Editing

The graphical editing is a manual process using GUI and a computer mouse. The process is intended for custom-designing a specific topology consisting of exact number of nodes located at exact locations and exact number of links. The design criteria are driven by the research needs. Editing changes that affect the currently displayed topology include reducing or increasing number of nodes, relocation of a node, and changes to the number of interconnection links. Changes made to the topology are recorded internally after the Update button is pressed.

3.4.1 Changing the Number of Nodes

Changes in the number of nodes include the addition of a new node and the deletion of an existing node, which are described in the following two subsections.

3.4.1.1 Node Addition

The creation of a new node is a manual process, which has been described in section 3.1.1. Every time a new node is created, it is added to the currently displayed topology. If the option to create a communication link is set to be dependent on the range of its radio, then all possible links between the newly created node and all other nodes are computed and displayed appropriately.

3.4.1.2 Node Deletion

The deletion of an existing node is also a manual process which can be done by right-clicking the target node to display a pop-up menu window from which the Delete Node menu item can be selected, as shown in figure 12. When this menu item is selected, the graphical representation of the target node is erased along with its communication links to other nodes if they exist. Then the target node is removed from the current topology.

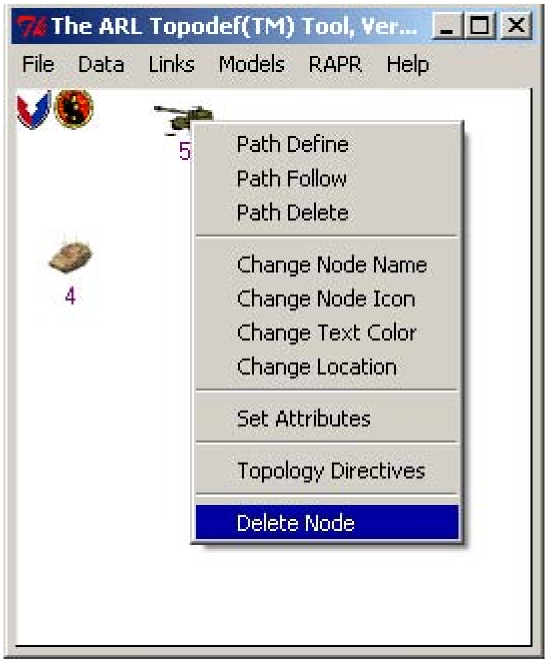


Figure 12. Node deletion.

3.4.1.3 Node Relocation

Node relocation is accomplished in the ARL Topodef tool by simply dragging the target node to a new position on the screen. The process affects the network topology being displayed on the screen because it changes the distances between the moving node and its neighbors. When the distances change, then the number of links may also change depending on how the option to create a link is set. If it is set to depend on a radio-signal propagation model, then all possible links between the moving node and all other nodes are computed and displayed appropriately. On the other hand, if it is set to be manual, then the number of links remains intact, although the graphical objects representing the links are stretched, shrunk, or rotated appropriately depending on the relative position of the moving node and its neighbors.

3.4.2 Changing the Number of Links

Changes to the number of links (link density) can be done by either creating a new link or removing an existing link. Changes made to the links of the currently displayed topology are reflected only in the generation of textual definition of a link between two nodes. Therefore, this feature is more useful to the definition of a network topology in the TEALab test bed than to the generation of *log* files.

3.4.2.1 Link Addition

The creation of a new link is done manually or automatically depending on how the link option is set. If it is set to be manually drawn, then a unidirectional or bidirectional link is created by dragging the cursor between two target nodes, as described in section 3.1.2. If the link option is set to be dependent on the free-space-loss model, then a new link is created by increasing the radio transmission power (power – tx), the reception sensitivity (power – rx), or the gain of the transmitter (gain – tx) or the receiver (gain – rx), as shown in figure 11.

3.4.2.2 Link Removal

The ARL Topodef tool provides two options to delete an existing link: using a pop-up menu or changing its radio parameters. The pop-up menu is displayed when the mouse cursor is resting on top of the target link and the right button of the mouse is pressed. The popup menu shows a list of menu items, among which is the Delete Link item (figure 13). If the existence of a link is set to depend on the free-space-loss model, then decreasing the radio transmission power (power – tx), the reception sensitivity (power – rx), or the gain of the transmitter (gain – tx) or the receiver (gain – rx) will delete the link (figure 11).

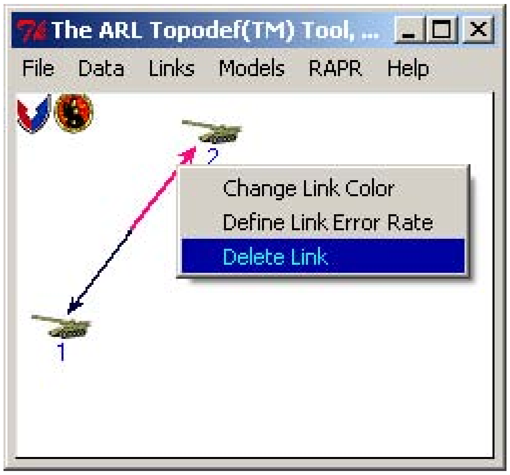


Figure 13. Removing a link originating from node 1 to node 2.

3.5 Scenario Manipulation

A mobility scenario is implemented internally as an ordered *list* of network topologies, as described in section 2.2. Scenario manipulation refers to the process of (1) re-arranging the order of the topologies in the scenario or (2) changing the length of the scenario (or equivalently changing the number of network topologies in the *list*).

Re-arranging the order of the topologies in the *list* is accomplished by “cutting” the currently displayed topology and “pasting” it at a different location in the scenario. This is also called permutation. The order of the topologies in an *n*-topology scenario can be re-arranged in *n!* different ways; i.e., an existing scenario can be used to create *n!* scenarios.

Changing the size of the scenario is accomplished by deleting the currently displayed topology or designing a new one and inserting it at the end of the scenario.

Figure 14 shows the appearance of two different toolsets. The left one is used for the manipulation of an existing scenario; the one on the right is for navigating and displaying a network topology in the scenario.

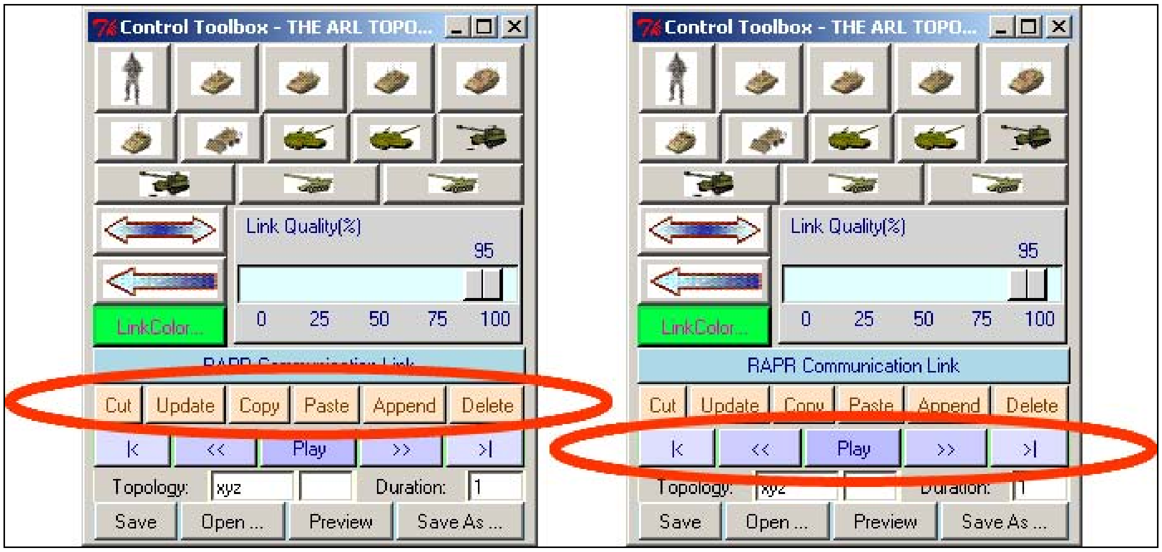


Figure 14. Graphical objects for scenario manipulation and navigation.

The descriptions of the encircled buttons on the right of figure 14 have been described in section 2.3. The paragraphs that follow explain the behavior and the implementation of the encircled buttons on the left.

The Cut Button. When this button is pressed, the specifications of the currently displayed topology are stored internally. Then the topology and its graphical representation are removed and erased; i.e., saved_topology=scenario.pop(i).

The Update Button. The graphical descriptions of currently displayed topology are updated internally whenever this button is pressed.

The Copy Button. This button is used to copy the specifications of the currently displayed topology into an internal storage; i.e., saved_topology=scenario[i].

The Paste Button. This button is used to insert the saved topology before the i^{th} slot of the internal *list*, which is the slot located before the ordinal number of the currently displayed topology; i.e., scenario.insert(i, saved_topology). The newly inserted topology is then displayed on the screen.

The Append Button. This button is used to insert the currently displayed topology into the end of the internal *list*.

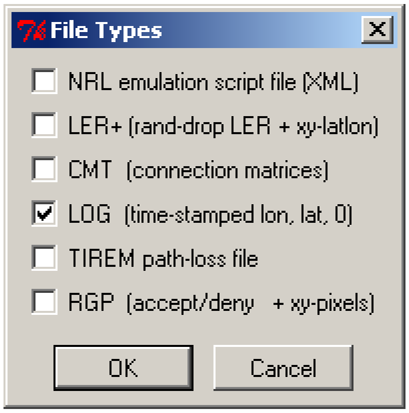
The Delete Button. This button is used to remove the currently displayed topology from the internal *list* and erases its corresponding graphical objects from the screen. The succeeding topology is then displayed on the screen if the *list* is not empty.

3.6 Input and Output Files

Input Files. A set of *log* files or a topology-definition file defines a mobility scenario. Either one can be loaded into the ARL Topodef tool for visualization, animation, or modification. The tool can also be used for converting the format between two different types of file; e.g., loading a set of log files, transforming them into a dynamic topology by applying a radio propagation model, and selecting a different type of file to save.

Output Files. The tool now offers the following types of output files, which are listed in a popup window when the option to save the scenario is chosen, as shown in figure 15:

- *NRL emulation script file*—A text file format defined by U.S. Naval Research Laboratory (NRL) to store geodesic positions of all participating mobile nodes.
- *LER+ file*—A type of topology-definition file intended for experimenting with link error rates (LERs) in the TEALab test bed. The file contains textual specifications of node positions and interconnection links. (The LERs are set to arbitrary values based on the distance between two nodes.)
- *CMT file*—A text file format for storing an ordered list of connection matrices to support link analysis. It is not used by any test bed.
- *LOG files*—A set of time-stamped geodesic positions of mobile nodes being emulated in the MANE test bed. Each participating node has a corresponding *log* file.
- *TIREM path-loss file*—An experimental binary file containing path-loss matrices. The path-loss computation employs the terrain integration rough earth model (TIREM) (*4*, *5*) and a DTED file. This option appears on the menu only if a DTED file and an appropriate version of the TIREM library are available on the host computer.
- *RGP file*—One of the earliest file formats defined by Richard Gopaul of ARL for specifying a dynamic topology in the TEALab test bed.



☐ NRL emulation script file (XML)
☐ LER+ (rand-drop LER + xy-latlon)
☐ CMT (connection matrices)
☑ LOG (time-stamped lon, lat, 0)
☐ TIREM path-loss file
☐ RGP (accept/deny + xy-pixels)

Figure 15. Output file types.

4. RAPR Links

A Real-Time Application Representative (RAPR) link is a graphical line connecting two communicating nodes—a server and a client—that use the RAPR software tool (*6*) to generate real-time traffic in the test bed. Therefore, RAPR should be running in every test node participating in a RAPR communication session.

Although RAPR links are not related to the definition of a network topology, they are provided by the tool as an add-on capability, a convenient feature for automatically generating script files. Each file defines a set of directives for requesting or generating a certain type of traffic at a specific time. The requester acts as the client, and the other, the server.

A RAPR link is created when the option to create a RAPR Communication Link is selected in the tool box and the mouse cursor is dragged between two nodes, from a client to a server. A RAPR link is represented on the screen as a dotted arrowed line originating from the client and ending at the server, as shown in figure 16.

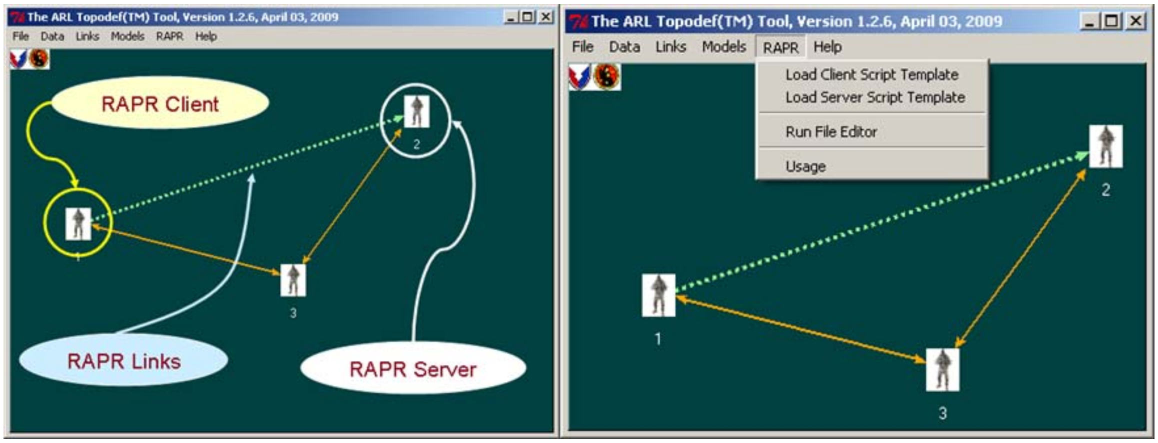
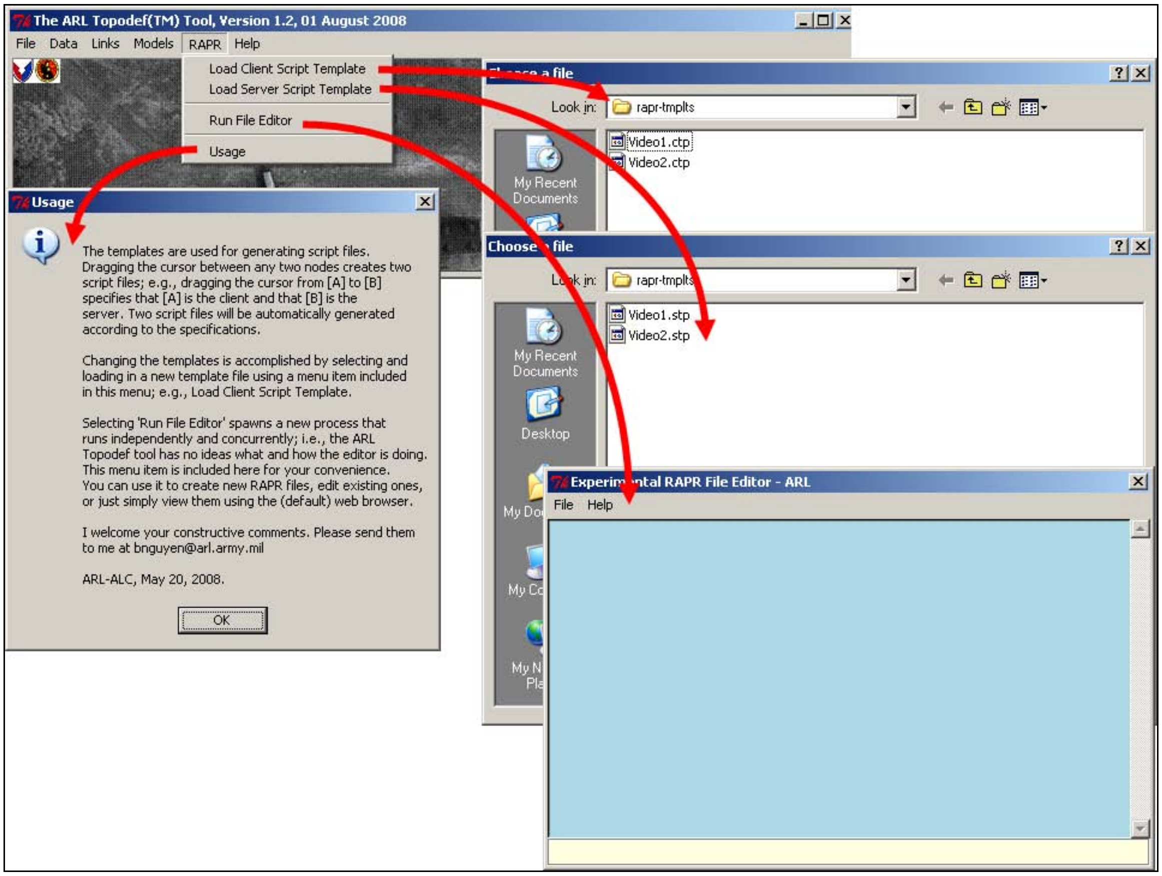

The ARL Topodef(TM) Tool, Version 1.2.6, April 03, 2009
File Data Links Models RAPR Help
RAPR Client
2
3
RAPR Links
RAPR Server
The ARL Topodef(TM) Tool, Version 1.2.6, April 03, 2009
File Data Links Models RAPR Help
Load Client Script Template
Load Server Script Template
Run File Editor
Usage
1
2
3

Figure 16. RAPR communication links.

After the graphical representation of a RAPR link is successfully drawn on the screen, two RAPR script files are generated and saved in the directory “rapr\scripts”; one is for the server and the other for the client, using their previously defined templates. Current implementation does not verify whether a RAPR communication session would be feasible or sustainable over a required period.

The templates are also script files with their parameters being substituted by the actual values used in the generation of script files. The tool uses a specific directory called “rapr\tmplts” to store RAPR template files whose names will appear in a popup window whenever the Load Client Script Template or the Load Server Script Template menu item is selected from the pull-down menu, as shown in figure 17. Furthermore, if the Run File Editor menu item is selected, then the “RaprEdt” tool (*7*), a graphical tool for creating RAPR files, is executed as an independent process.



Usage

The templates are used for generating script files. Dragging the cursor between any two nodes creates two script files; e.g., dragging the cursor from [A] to [B] specifies that [A] is the client and that [B] is the server. Two script files will be automatically generated according to the specifications.

Changing the templates is accomplished by selecting and loading in a new template file using a menu item included in this menu; e.g., Load Client Script Template.

Selecting 'Run File Editor' spawns a new process that runs independently and concurrently; i.e., the ARL Topodef tool has no ideas what and how the editor is doing. This menu item is included here for your convenience. You can use it to create new RAPR files, edit existing ones, or just simply view them using the (default) web browser.

I welcome your constructive comments. Please send them to me at bnguyen@arl.army.mil

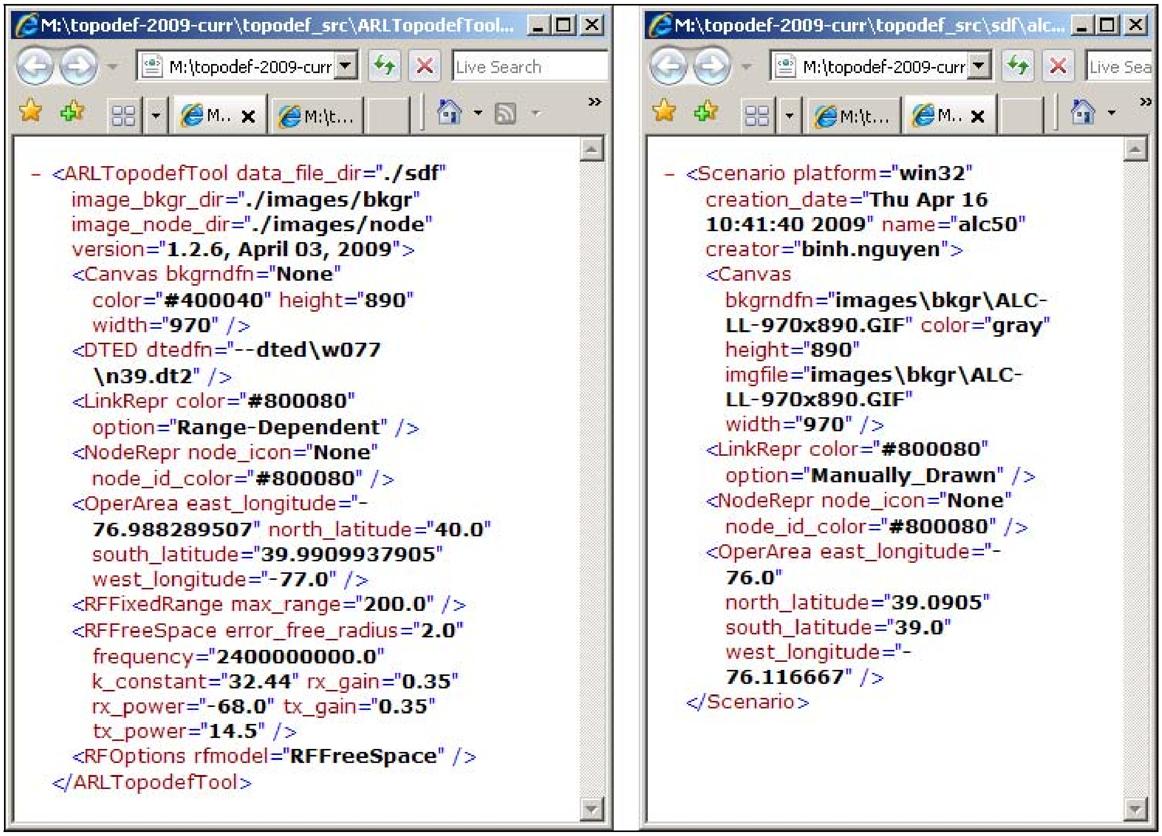
ARL-ALC, May 20, 2008.

OK

Figure 17. The RAPR pull-down menu.

5. Configuration Files

Two different types of configuration files are used by the ARL Topodef tool; they are not needed for emulation purposes. One is used to configure the ARL Topodef tool, and the other to provide information about an existing scenario; respectively, they are named “ARLTopodefToolConfig.xml” and “ARLTopodefScenario.xml” and are shown in figure 18 from left to right.


M:\topodef-2009-curr\topodef_src\ARLTopodefTool...
M:\topodef-2009-curr
Live Search
M..
M:\t...
M:\topodef-2009-curr\topodef_src\sdf\alc...
M:\topodef-2009-curr
Live Sea
M:\t...
M..

```
- <ARLTopodefTool data_file_dir="./sdf"
    image_bkgr_dir="./images/bkgr"
    image_node_dir="./images/node"
    version="1.2.6, April 03, 2009">
    <Canvas bkgrndfn="None"
      color="#400040" height="890"
      width="970" />
    <DTED dtedfn="--dted\w077
      \n39.dt2" />
    <LinkRepr color="#800080"
      option="Range-Dependent" />
    <NodeRepr node_icon="None"
      node_id_color="#800080" />
    <OperArea east_longitude="-
      76.988289507" north_latitude="40.0"
      south_latitude="39.9909937905"
      west_longitude="-77.0" />
    <RFFixedRange max_range="200.0" />
    <RFFreeSpace error_free_radius="2.0"
      frequency="2400000000.0"
      k_constant="32.44" rx_gain="0.35"
      rx_power="-68.0" tx_gain="0.35"
      tx_power="14.5" />
    <RFOptions rfmodel="RFFreeSpace" />
  </ARLTopodefTool>
```

```
- <Scenario platform="win32"
    creation_date="Thu Apr 16
    10:41:40 2009" name="alc50"
    creator="binh.nguyen">
    <Canvas
      bkgrndfn="images\bkgr\ALC-
      LL-970x890.GIF" color="gray"
      height="890"
      imgfile="images\bkgr\ALC-
      LL-970x890.GIF"
      width="970" />
    <LinkRepr color="#800080"
      option="Manually_Drawn" />
    <NodeRepr node_icon="None"
      node_id_color="#800080" />
    <OperArea east_longitude="-
      76.0"
      north_latitude="39.0905"
      south_latitude="39.0"
      west_longitude="-
      76.116667" />
  </Scenario>
```

Figure 18. Contents of configuration files being shown in a web browser.

The configuration parameters are organized in the files as eXtensible Markup Language (XML) tags; the files are, thus, XML files, which can be viewed using a web browser and edited using a text editor. Each tag has a name and a list of attributes.

The ARLTopodefToolConfig.xml file is automatically created by the ARL Topodef tool. If the file exists, then its contents are used to configure the tool at the start of its execution and updated at the end. Among the many XML tags in the file, the DTED and the OperArea tags are the two most important ones. The DTED tag specifies the name of a DTED file. If the specified DTED file is found, then its embedded information about its bounding rectangle overrides the specifications in the OperArea tag.

The ARLTopodefScenario.xml file is also automatically created by the ARL Topodef tool whenever a mobility scenario is saved. This file is used by the tool for visualization and animation purposes. If it does not exist, then the input data may not be correctly displayed on the screen. The most important tag in the file is the OperArea tag because it specifies the geodetic boundaries of the scenario. Changes to the boundary values produce zooming effect when the scenario is visualized.

6. Summary

The ARL Topodef tool is a graphical system and method for creating, among other things, virtual network topologies to drive network emulation in the MANE or the TEALab test beds. The tool uses mobility models and propagation models, similar to but distinguishable from a traditional modeling or simulation tool; e.g., Opnet[1] or Qualnet[2]. The models are used to provide the user a way to experiment with the model parameters to provide a convenient way to create a dynamic topology expeditiously, and to graphically alter it in order to fit a specific research need.

The tool offers its users two different ways to create a desired mobility scenario. The first approach accomplishes the objective revolutionarily by completing the mobility data-generation process in reverse order: (1) first, graphically designing a desired dynamic topology, and then (2) generating the mobility-pattern data corresponding to the design. The second approach accomplishes the objective evolutionarily by (1) simultaneously displaying and generating a dynamic scenario, (2) visually editing and manipulating the graphical elements depicting a topology to produce a desired result, and then (3) generating the mobility-pattern data corresponding to the design.

The tool stores the specifications of each topology in a mutable composite data structure called a *list*, which is an ordered collection of topology specifications. The *list* can be embodied as an electronic flipbook in which every page has the required specifications of a topology. Each topology is associated with an ordinal number called an index, indicating its ordered position within the list. The index is used to select a topology in the list. Once a topology is saved in the list, it can be retrieved, modified, updated, removed from the list, or relocated to a different location within the list. An n-topology list can be permuted to create n-factorial (n!) different scenarios.

The tool provides a method for graphically animating and editing mobility traces of a MANET, and for transforming them into a dynamic topology. The employed mobility traces comprise time-stamped positions of each participating mobile node, and can be structured differently in diverse file formats and from different sources.

The tool is also used to load externally generated mobility traces to serve three major research purposes: (1) providing graphical aid in understanding movement patterns, (2) animating a dynamic network topology, and (3) creating derivatives of the existing traces to fit a particular need, effectively enabling the user to leverage existing resources by reusing mobility traces.

[1] Opnet is a commercial product of OPNET Technologies, Inc.

[2] Qualnet is a commercial product of Scalable Network Technologies.

Thus, the tool reduces internal development efforts and enables the try-out of mobility scenarios developed by others more conveniently.

The tool will be improved to meet future research needs. Plans for enhancing the tools include (1) generating mobility scenarios and path-loss files in various formats to increase its portability, (2) developing efficient storage and processing techniques for working with a large MANET, (3) using multiple radios operating at different frequencies or equipped with directional antennas or both, (4) integrating group mobility models, (5) improving graphical user interfaces and visual appearances, and (6) enabling the tool to broadcast or to receive GPS coordinates in the test bed.

7. References

1. Corson, S.; Macker, J. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC# 2501, January 1999. http://www.ietf.org/rfc/rfc2501.txt (accessed April 24, 2009).

2. Little, M. TEALab: A Testbed for Ad-Hoc Networking Security Research, Military Communications Conference, 2005. MILCOM 2005. IEEE, 17-20 Oct. 2005

3. Network and Communication Systems Branch, Mobile Ad-hoc Network Emulator (MANE), U.S. Naval Research Laboratory (NRL), Washington, D.C. http://cs.itd.nrl.navy.mil/work/mane/ (accessed April 24, 2009).

4. Eppink, D.; Kuebler, W. TIREM/SEM Handbook, ECAC-HDBK-93-076, Department of Defense Electromagnetic Compatibility Analysis Center, Annapolis, Maryland 21402-5064, March 1994. URL: http://handle.dtic.mil/100.2/ADA296913 (accessed 07 January 2009).

5. TIREM, a commercial product of Alion Science and Technology, URL: http://www.alionscience.com (accessed 05 October 2009).

6. Networks and Communications Systems Branch, “RAPR - The Real-Time Application Representative, The U.S. Naval Research Laboratory, Code 5520, 4555 Overlook Ave., SW, Washington, DC 20375-5337. http://cs.itd.nrl.navy.mil/work/rapr (accessed 01 May 2009).

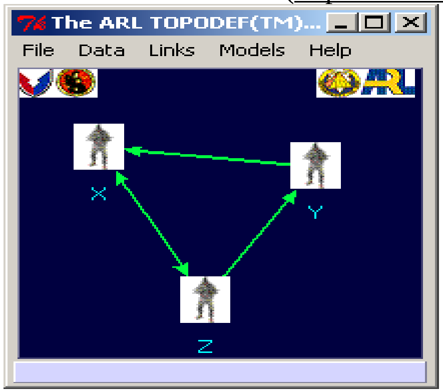
7. Nguyen, Binh Q. *The ARL RaprEdt Tool—A Graphical Editor for Creating Real-time Application Representative (RAPR) Files*; ARL-TR-4600; U.S. Army Research Laboratory: Adelphi, Maryland, September 2008.

Appendix. Terminologies

MANET

Mobile ad hoc network (http://www.ietf.org/rfc/rfc2501.txt)

Graphical network topology



Textual Definition of a Network Topology (topology definition)

```
on  X position 44.452973 33.3291647807
on  Y position 44.453074 33.3291648000
on  Z position 44.452990 33.3109147807
on  X accept Y inbound
on  X accept Z inbound
on  Y accept Z inbound
on  Y deny X inbound
on  Z accept X inbound
on  Z deny Y inbound
wait for 1 second
```

Textual definitions of a dynamic topology

A time-ordered list of discrete network-topology definitions

Scenario

A dynamic topology or a mobility scenario

RF

Radio frequency

Range model

A mathematical model that computes the transmission range of a radio

FSL model

Free Space Loss model – an RF signal propagation model

The ARL Topodef Tool

A graphical tool and method that was developed by the U.S. Army Research Laboratory (ARL) to support MANET emulation. The tool serves multiple purposes:

(1) design, visualization, and animation of a network topology
(2) modification, visualization, and animation of mobility traces
(3) automatic creation of textual network-topology definitions
(4) generation of path-loss matrices
(5) demonstration of the dynamic topology of a MANET

List of Symbols, Abbreviations, and Acronyms

ARL	U.S. Army Research Laboratory
DTED	digital terrain elevation data
GIF	graphics interchange format
GUIs	graphical user interfaces
LERs	link error rates
MANET	mobile ad-hoc network
NRL	Naval Research Laboratory
OS	operating system
RAPR	Real-Time Application Representative
TDF	topology-definition file
TIREM	terrain integration rough earth model
TNs	test nodes
TSM	topology-scenario manager
XML	eXtensible Markup Language

NO. OF
COPIES ORGANIZATION

1 ADMNSTR
ELEC DEFNS TECHL INFO CTR
ATTN DTIC OCP
8725 JOHN J KINGMAN RD STE 0944
FT BELVOIR VA 22060-6218

1 DARPA
ATTN IXO S WELBY
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

1 CD OFC OF THE SECY OF DEFNS
ATTN ODDRE (R&AT)
THE PENTAGON
WASHINGTON DC 20301-3080

1 US ARMY RSRCH DEV AND ENGRG CMND
ARMAMENT RSRCH DEV AND ENGRG CTR
ARMAMENT ENGRG AND TECHNLGY CTR
ATTN AMSRD AAR AEF T J MATTS
BLDG 305
ABERDEEN PROVING GROUND MD 21005-5001

1 PM TIMS, PROFILER (MMS-P) AN/TMQ-52
ATTN B GRIFFIES
BUILDING 563
FT MONMOUTH NJ 07703

1 COMMANDER
US ARMY RDECOM
ATTN AMSRD AMR W C MCCORKLE
5400 FOWLER RD
REDSTONE ARSENAL AL 35898-5000

1 US GOVERNMENT PRINT OFF
DEPOSITORY RECEIVING SECTION
ATTN MAIL STOP IDAD J TATE
732 NORTH CAPITOL ST NW
WASHINGTON DC 20402

1 US ARMY RSRCH LAB
ATTN RDRL CIM G T LANDFRIED
BLDG 4600
ABERDEEN PROVING GROUND MD 21005-5066

1 BINH NGUYEN (B.NGUYEN@IEEE.ORG) 1 E-COPY
ELEC

3 US ARMY RSRCH LAB
ATTN IMNE ALC HRR MAIL & RECORDS MGMT
ATTN RDRL CIM L TECHL LIB
ATTN RDRL CIM P TECHL PUB
ADELPHI MD 20783-1197

TOTAL: 12 (2 ELEC, 1 CD, 9 HCS)

INTENTIONALLY LEFT BLANK.