Army Research Laboratory



The ARL RaprEdt Tool—A Graphical Editor for Creating Real-time Application Representative (RAPR) Files

by Binh Q. Nguyen

ARL-TR-4600

September 2008

Approved for public release; distribution unlimited.

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

September 2008

The ARL RaprEdt Tool—A Graphical Editor for Creating Real-time Application Representative (RAPR) Files

Binh Q. Nguyen Computational and Information Sciences Directorate, ARL

Approved for public release; distribution unlimited.

	REPORT DO	DCUMENTAT	ION PAGE		Form Approved OMB No. 0704-0188		
Public reporting bur existing data source burden estimate or a Headquarters Servic 4302. Respondents collection of inform	den for this collection s, gathering and maint uny other aspect of this ces, Directorate for Info should be aware that r ation if it does not disp	of information is estin aining the data needed collection of informa ormation Operations a notwithstanding any of play a currently valid	mated to average 1 hd 1, and completing and ttion, including sugge and Reports (0704-01 ther provision of law, OMB control number	our per response, incl l reviewing the colle- stions for reducing the 88), 1215 Jefferson I no person shall be s	luding the time for reviewing instructions, searching ction information. Send comments regarding this he burden, to Department of Defense, Washington Davis Highway, Suite 1204, Arlington, VA 22202- subject to any penalty for failing to comply with a		
1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE			3. DATES COVERED (From - To)		
Septemb	per 2008		Final		Fiscal Year 2008		
4. TITLE AND SUBT	ITLE				5a. CONTRACT NUMBER		
The ARL RaprE	Edt Tool—A Grap	hical Editor for C	reating Real-time	e Application			
Representative	(RAPR) Files				5b. GRANT NUMBER		
					5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)					5d. PROJECT NUMBER		
Binh Q. Nguyer	1				5e. TASK NUMBER		
					5f. WORK UNIT NUMBER		
7. PERFORMING O	RGANIZATION NAME	S) AND ADDRESS(E	S)		8. PERFORMING ORGANIZATION		
U.S. Army Rese	earch Laboratory				REPORT NUMBER		
ATTN: AMSR	D-ARL-CI-NT				ARL-TR-4600		
2800 Powder M	ill Road						
Adelphi, MD 20)/83-119/ ONITORING AGENCY	NAME(S) AND ADDR	(FSS(FS)		10. SPONSOR/MONITOR'S ACRONYM(S)		
			.200(20)				
					11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/	AVAILABILITY STATE	MENT					
Approve for pul	olic release; distrib	oution unlimited.					
13. SUPPLEMENTA	RY NOTES						
14. ABSTRACT							
The U.S. Army Research Laboratory (ARL) designed and developed a tool called the ARL Real-time Application Representative (RaprEdt) tool to support the emulation of communication scenarios by providing a method for rapidly creating critical files that are required for using the RAPR Version 1.0, a product of the U.S. Naval Research Laboratory (NRL). This report documents the functional behavior of the ARL RaprEdt tool and describes its graphical user interfaces (GUIs).							
15. SUBJECT TERM	IS						
RAPR, editor, C	GUI						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF	18. NUMBER OF	19a. NAME OF RESPONSIBLE PERSON Binh Q. Nguyen		
a. REPORT	b. ABSTRACT	c. THIS PAGE		PAGES	19b. TELEPHONE NUMBER (Include area code)		
U	U	U		20	(301) 394-1781		

Standard Form 298 (Rev. 8/98) Prescribed by ANSI Std. Z39.18

Contents

Co	ontents	iii
Lis	st of Figures	iv
Su	mmary	1
1.	Background	3
2.	The ARL RaprEdt Tool	4
	2.1 The RAPR Dictionary Editor	6
	2.2 The RAPR Logic-Table Editor	7
	2.3 The RAPR Script Editor	8
3.	Conclusion	12
Ac	ronyms	13
Dis	stribution List	12

List of Figures

Figure 1.	Appearance and usage instructions of the RaprEdt tool	5
Figure 2.	Selecting a type of RAPR file.	6
Figure 3.	Editing a RAPR dictionary	7
Figure 4.	Editing a RAPR logic table.	8
Figure 5.	Editing a RAPR script by selecting an MGEN pattern.	9
Figure 6.	List of reception events and global commands.	10
Figure 7.	List of RAPR behavior events.	11
Figure 8.	List of RAPR events, MGEN patterns, and run-time interfaces.	12

Summary

The Real-time Application Representative (RAPR) Version 1.0 is a product of the U.S. Naval Research Laboratory (NRL). It is designed mainly to generate communication traffic. Its behavior is defined by the textual specifications stored in three different types of computer files: (1) a dictionary file, (2) a logic-table file, and (3) script files. The three file types are collectively called RAPR files. The files are used to specify a communication scenario in a laboratory environment. They define the role and the behavior of each participating computer in a network test bed.

Experimenting with various communication scenarios in a test bed requires distinguishable RAPR files for each scenario. Each scenario involves different number of communicating entities. When the number of communicating entities increases, the manual creation of the files would be a tedious, error-prone, and time-consuming process. Circumventing the manual process necessitate the development of an automated tool called "the U.S. Army Research Laboratory (ARL) RaprFileEDT tool" capable of generating RAPR files more rapidly.

Learning how to use the ARL RaprEdt tool requires very little time because it is equipped with graphical user interfaces with which the user is very familiar. Using the tool, a communication engineer can focus on the development of a communication scenario without being concerned with the mechanical structure of the files.

INTENTIONALLY LEFT BLANK.

1. Background

The U.S. Army Research Laboratory (ARL) uses the Real-time Application Representative¹ (RAPR) Version 1.0 to generate and response to communication traffic and events in the wireless emulation laboratory (WEL), which has been designed and constructed to emulate a dynamic movement of a mobile ad hoc network (MANET). It is a high-performance test bed consisting of a gigabit network connecting 48 physical computers. Each computer is capable of hosting a sub-network of virtual machines (VMs) loaded with a version of the Linux® operating system to host advanced MANET routing protocols, networked applications, information-assurance (IA) products, and performance measurement and visualization tools. The WEL test bed is also a showcase for ARL innovation and successes.

The U.S. Naval Research Laboratory (NRL) designs and develops the RAPR program for experimentation purposes. The RAPR program is used to generate communication traffic and to measure and capture its performance metrics for subsequent analysis. The behavior of a communication scenario is defined by the contents of the files that are associated with a running RAPR program and that are designed specifically for a particular scenario. The files are collectively called RAPR files consisting of dictionary, logic table, and script files.

Experimenting with various communication scenarios in the ARL WEL test bed requires a set of RAPR files for each scenario. Each scenario involves different number of communicating entities. Some act as servers and some act as clients. The RAPR files are used to define the roles and their related behavior of the participating MANET nodes. The number of mobile nodes that can be emulated in the WEL test bed is increasing rapidly. The WEL test bed now can accommodate the emulation of a 100-node MANET, and it is being improved to handle a 1000-node MANET.

Because the manual creation of RAPR files for large networks consisting of hundreds of nodes would be a tedious and time-consuming process, ARL requires that the process be augmented with an automated tool capable of facilitating and expediting the process. In fiscal year 2008 (FY08), ARL conducted a requirement analysis that resulted in a set of performance specifications for an automated tool² that would enable the creation of RAPR files more rapidly by freeing its users from being concerned with the required detailed structure of the files. The specifications were then used to develop the ARL RaprEdt tool, also known as the ARL RaprFileEDT, which is reported in this report.

¹ 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. <u>http://cs.itd.nrl.navy.mil/work/rapr</u> (accessed 05 November 2007).

² Hardy, R.; Nguyen, B. ,*Performance Requirements of Tools and Methods for Specifying Network Communication Scenarios Using the Real-Time Application Representative Version 1.0*; ARL-TR-4614; U.S. Army Research Laboratory: Adelphi, MD 20783, September 2008.

Subsequent sections of this report describe the ARL RaprEdt tool and include the screenshots illustrating the features that are built into the tool. The final section concludes the report with a plan for integrating the tool with other ARL-developed tools to support the development, test, and evaluation of MANET technologies and applications.

2. The ARL RaprEdt Tool

The RaprEdt tool is an editing tool equipped with graphical user interface (GUI) features. The tool was created using the Python programming language and its built-in Tkinter module³. The tool is an integrated tool comprising three editing tools: (1) the dictionary editor, (2) the logic-table editor, and (3) the script editor. The three editing tools share the main GUI, which is used for selecting a file type, saving and viewing a file, and observing the results of an editing session.

Figure 1 shows the initial appearance of the tool and its usage instructions, which are displayed whenever a menu item in the **Help** menu is pulled down and selected. This is the first step a new user would do before using the tool to create a RAPR file.

The **File** menu of the tool, being placed at the upper-left corner of the screen, has a list of menu items that deal with a RAPR file. This section describes the behavior of three commonly performed actions: creating a new file or editing and viewing an existing one. Other menu items, such as the **Save** operations, are not described in this report because their functional behavior is self-explanatory and similar to many other software applications.

³ The Python Programming Language. <u>http://www.python.org</u> (accessed 08 July 2008).



Figure 1. Appearance and usage instructions of the RaprEdt tool.

Figure 2 shows the screenshots when a **File** menu item is selected and the ensuing actions that the user needs to perform. To create a new RAPR file, the user needs to select the menu item labeled **New** and specifies a file type. When the **New** menu item is selected, the tool opens a dialog window displaying three radio buttons indicating the three different file types: dictionary, logic table, and script. The user then selects a file type by clicking an appropriate radio button and the **OK** button. The first two file types contain the text being marked up using the extensible markup language (XML), and the last type is intended for storing a sequence of RAPR scripts, consisting of timed events and commands.

Selecting the menu item **Open** or **View** opens a file-dialog window showing a list of existing XML files from which the user can choose one. If the **Open** operation is selected, then the contents of the selected XML file are loaded into the tool, which automatically determines its file type, i.e., whether it is a dictionary or a logic-table file. If the **View** operation is selected, then the contents of the selected XML file are loaded and displayed in the default Web browser of the host system on which the tool runs. Figure 2 shows the Microsoft Internet Explorer displaying the contents of a RAPR dictionary file.



Figure 2. Selecting a type of RAPR file.

2.1 The RAPR Dictionary Editor

A RAPR dictionary contains a set of unique keys or names that are used to look up their associated values. A RAPR dictionary is used to translate name-value pairs used in RAPR files. Each dictionary has one or more **namespace** fields. Each namespace has a **label** field and one or more **item** fields. Each field is given a unique name and assigned one or more values.

When a RAPR dictionary is being edited, the ARL RaprEdt tool displays a set of editing options that are specifically designed for editing a RAPR dictionary. Figure 3 displays the screenshot of the ARL RaprEdt tool when it deals with a RAPR dictionary. The rules for creating a dictionary and the XML tags are shown to its users for informational purposes only, i.e., the user does not need to deal with the XML tags. Figure 3 also shows two examples. The first example illustrates the entry of single name-value pair into a dictionary, and the second the entry of multiple name-value pairs, which is specified in a single line within the field entry box. Note that the entries in the dictionary are not alphabetically sorted.



Figure 3. Editing a RAPR dictionary.

2.2 The RAPR Logic-Table Editor

A RAPR logic table describes event-driven behavior of a communication scenario. Each table consists of one or more states. Each state has one or more logicids. The data structure of each logicid has an **id** field, a **percent** field, and one or more **entry** fields. Each entry defines the behavior associated with a given logicid. The **percent** field defines the probability of the triggering event, and it ranges from 0.0 to 1.0.

When a RAPR logic table is to be edited, the ARL RaprEdt tool displays a set of editing options that are specifically designed for editing a logic table. Figure 4 displays the screenshot of the ARL RaprEdt when it deals with a RAPR logic table. The rule for creating a logic table and the XML tags are shown to its users for informational purposes only, i.e., the user does not need to deal with the XML tags. The contents of the field entry box can be manually typed in or left blank. When it is left blank, clicking the **Add/Edit** button will display a window showing the various events and commands from which the user can select. The procedure for completing the process is similar to the process of creating a RAPR script file, which is described in section 2.3, the RAPR Script section.

			74 ARL::RaprFileED	T - a method for creati	ng RAPR files				- 🗆 🗵
			File Help						
			value	alpha				Add/Update	Remove
			id	two				Add/Update	Remove
74 ARL:RAP	R Files		percent	.70				Add/Update	Remove
O D	ictionary		entry	192.101.102.111				Add/Edit	Remove
• L O S	ogicTable criptFile		<raprlogictable> <!--The table has one</td--><td>or more states. Each state I</td><td>nas one or more</td><td>logic IDs.</td><td></td><td></td><td><u> </u></td></raprlogictable>	or more states. Each state I	nas one or more	logic IDs.			<u> </u>
	1		Each logic ID has a pe <state></state>	ercent field (default=1.0) an	d one or more er	ntries>			
ОК	Cancel		<value>alpha<td>alue></td><td>ARL::RAPR F</td><td>ile Editor</td><td></td><td></td><td>×</td></value>	alue>	ARL::RAPR F	ile Editor			×
			<id>two<td>> 70</td><td>The t</td><td>able bas on</td><td>e or more states. Each state bas</td><td>one or more logi</td><td>r IDs.</td></id>	> 70	The t	able bas on	e or more states. Each state bas	one or more logi	r IDs.
			<pre> <pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>	32.101.102.111	Each	logic ID has	a percent field (default=1.0) an	id one or more en	tries.
			<logicid></logicid>						
76 ARL::RaprFileED	T - a method fo	r creating RAPR fil	es				L		
File Help									
value	beta				Add/Update	Remove			
id	two				Add/Update	Remove			
percent	0.8 0				Add/Update	Remove			-
entry					Add/Edit	Remove			
<pre><raprlogictable> <!---The table has one or more states. Each state has one or more logic IDs. Each logic ID has a percent field (default=1.0) and one or more entries--> <!--/RaprLogicTable--> </raprlogictable></pre>									
		74 ARL::RAPR Logi	Table Entries						×
		C Reception Ever	nts – C. Behavior Ev	vents C RAPR Events	🖌 🔿 MGEN	Patterns	C Run-Time Interface C	RAPR Global Co	mmands
								Keep Cano	cel Close

Figure 4. Editing a RAPR logic table.

2.3 The RAPR Script Editor

The RAPR script specifies an event, a Multi-Generator (MGEN) pattern, run-time interface, or RAPR command. An event can be either a reception event, a behavior event, or a RAPR event. The RAPR script editor provides the user two different ways to create a RAPR script: (1) manually typing in a script or (2) using a series of pop-up windows to select an appropriate script and to set the values of its parameters if it requires.

Clicking the **Add/Edit** button while the **Command** field is empty displays a window showing the various events and commands from which the user can select. Figure 5 shows the MGEN Patterns option and its POISSON pattern, among the three patterns, are selected. Once an option is surely chosen, the user has to press the **Keep** button to retain the option. The RAPR script editor tracks a list of selected options that the user wants to keep. Figures 6–8 show various ways for creating a RAPR script by specifying a reception event, a global command, a behavior command, a RAPR event, or a run-time interface directive.

ARLISRAPE File File Help Comment Comment Example 1: Join subnet abo at 32.45 Add/Update Remove OK Carce HE sample 1: Join subnet abo at 32.45 ARLISRAPE File Editor Idd/Edit Memove OK Carce HE sample 1: Join subnet abo at 32.45 ARLISRAPE File Editor Idd/Edit Idd/Edit <td< th=""><th></th><th>76 ARL::RaprFileED</th><th>T - a method for creatir</th><th>ig RAPR files</th><th></th><th></th><th></th><th></th></td<>		76 ARL::RaprFileED	T - a method for creatir	ig RAPR files				
Comment Example 2. Join subnet abo at 32.45 Add/Update Remove Comment Example 2. Join subnet abo at 32.45 Add/Update Remove Comment Example 2. Join subnet abo at 32.45 Add/Update Remove REsample 2. Join subnet abo at 32.45 Add/Update Remove REsample 2. Join subnet abo at 32.45 Add/Update Remove REsample 2. Join subnet abo at 32.45 Add/Update Remove REsample 2. Join subnet abo at 32.45 Add/Update Remove REsample 2. Join subnet abo at 32.45 Add/Update Remove Reception Events RAPR Events RAPR Events Remove Reception Events RAPR Events RAPR Events Remove Reception Events Rehavior Events RAPR Events RAPR Events Remove Reception Events Rehavior Events RAPR Events RAPR Events Remove Reception Events Rehavior Events RAPR Events Remove Reception Events Rehavior Events RAPR Events Remove Remove Remove Remove Reception Events Rehavior Events RAPR Events Remove Remov	7. ARL::RAPR Files X	File Help						
Add/Edit Remove Scipifie Command 32 45 JOIN abc Add/Edit Remove HEsample 1: Join submet size at 12 40 12 40 JOIN ye HEsample 2: Join submet size at 12 40 12 40 JOIN ye HEsample 2: Join submet abc at 32.45 Scipifie Cancel Cance	O Dictionary	Comment	Example 2: Join subnet ab	c at 32.45			Add/Up	date Remove
Comment ARL::RAPR FileEDI - a method for creating RAPR files File Hep Comment Add/Lipdate Reception Events Reception Events <	C LogicTable	Command	32.45 JOIN abc				Add/E	dit Remove
DK Cance If Cooperation system If Cooperation system Sequence of commands, scheduled events, and other options. If ARL:RAPF FileEDT - a method for creating RAPR files Image: Comment in the image: Commands in the image: Command in the ima	ScriptFile	# Example 1: Join sub 12:40-101NL www	net xyz at 12.40					<u>^</u>
ARL:RAPF Script ARL:R	OK Cance	# Example 2: Join sub	net abc at 32,45	76 ARL::RAPR	File Editor			×
ARL:RAPF Script ARL:R		32.45 JUIN abc		()				-Nora
ARL:RAPF Script OK File Help Comment Add/Update Comment Add/Edit Reception Events Behavior Events Reception Events					quence or command:	s, scheduled eve	ents, and other o	ptions.
File Help Comment Add/Update Remove Command Add/Update Remove Add/Edit Remove ARL:RAPR Script X Reception Events Behavior Events RAPR Events Run-Time Interface RAPR Global Commands Keep Cancel Close X Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands PERIODIC FERIODIC[(rateo < size)]	ARL::RaprFileEDT - a method for creating RAPR f	iles			[[OK		
Comment Add/Update Remove Command Add/Cdt Remove ARL:RAPR Script Image: Command Stript Image: Command Stript ARL:RAPR Script Image: Command Stript Image: Command Stript ARL:RAPR Script Image: Command Stript Image: Command Stript Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands PERIODIC PERIODIC(rate> <size)< td=""> rate zvooig size (types) size (types) POISSON POISSON POISSON(aveRate (msg/sec) <size (bytes))<="" td=""> aveRate (msg/sec) 1024 size (types) size (types) BURST BURST(REGULARIRANDOM caveInterval (sec)) aveDateIntraje (sec) 10 patternParams (syegodn reVEDEXPONENTIAL <aveduration (sec))<="" td=""> aveDuration (sec) 10 POISSON(1024 4096] Keep Cancel Close</aveduration></size></size)<>	File Help							
Command Add/Edit Renove ARL::RAPR Script X Reception Events Behavior Events RAPR Events Run-Time Interface RAPR Global Commands Keep Cancel Close Close X Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands PERIODIC PERIODIC (rate> <size)< td=""> rate zvocity size (bytes) size (bytes) size (bytes) BURST BURST[REGULAR RANDOM <aveinterval (sec)=""> (patternParams)] aveInterval (sec) 10 patternType (jozi patternParams cryegovin FXEDEXPONENTIAL <aveduration (sec))<="" td=""> aveDuration (sec) 10 patternType (jozi patternParams cryegovin FXEDEXPONENTIAL <aveduration (sec))<="" td=""> aveDuration (sec) 10 patternType (jozi patternParams cryegovin</aveduration></aveduration></aveinterval></size)<>	Comment			Add/Update F	Remove			
ARL::RAPR Script Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Keep Cancel Close Keep Cancel Close Reception Events Behavior Events RAPR Revents RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands PERIODIC (rate> <size) (bytes)="" (msg="" (sec)="" <aveduration="" aveduration="" aveinterval="" averate="" cancel="" close<="" cryegxdn="" fixediexponential="" keep="" patternparams="" qvurz="" rate="" sec)="" size="" td="" to="" to24="" zvooi=""><td>Command</td><td></td><td></td><td>Add/Edit F</td><td>Remove</td><td></td><td></td><td></td></size)>	Command			Add/Edit F	Remove			
ARL::RAPR Script Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Keep Cancel Close Close Close Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands rete zvooiy size qvurz eptionEvents Behavior Events RAPR Global Commands rete zvooiy size qvurz eptionEvents Behavior Events RAPR Global Commands rete zvooiy size qvurz eptionEvents BunstT BURST[REGULARIRANDOM (aveInterval (sec)> (patternType) (cpatternParams)] aveInterval (sec) 10 patternType glozi potstorn(1024 4096] Keep Cancel Close				/				
ARL:RAPR Script Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Keep Cancel Close Keep Cancel Close Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands PERIODIC (rate> <size)< td=""><td></td><td></td><td>/</td><td></td><td></td><td></td><td></td><td></td></size)<>			/					
Reception Events Behavior Events RAPR Events MGEN Patterns RAPR Global Commands Keep Cancel Close Keep Cancel Close Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands PERIODIC PERIODIC(vate> <size) (bytes)="" (bytes))="" (msg="" (sec)="" 1024="" 4094="" <aveinterval="" <size="" averate="" burst="" burst[regular random="" poisson="" poisson(<averate="" sec)="" size=""> (patternType) (cpatternParams)] aveInterval (sec) 10 POISSON(1024 4096] Keep Cancel Close Close</size)>	7% ARL::RAPR Script						×	-
Keep Cancel Close ** ARL:RAPR Script ** ** Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands ** PERIODIC PERIODIC(rate> <size)< td=""> rate zvoojy size guurz ** POISSON POISSON(aveRate (msg/sec) < size (bytes)>) aveRate (msg/sec) 1024 size (bytes) 4094 ** BURST BURST[REGULAR]RANDOM <aveinterval (sec)=""> (patternType) <patternparams)< td=""> aveInterval (sec) 10 patternParams cryegxdn FXEDIEXPONENTIAL <aveduration (sec)=""> aveDuration (sec) 10 patternParams cryegxdn FVEDIEXPONENTIAL <aveduration (sec)=""> aveDuration (sec) 10 patternParams cryegxdn FVEDIEXPONENTIAL <aveduration (sec)=""> aveDuration (sec) 10 patternParams cryegxdn</aveduration></aveduration></aveduration></patternparams)<></aveinterval></size)<>	C Reception Events C Behavior Even	ts C RAPR Event:	s C MGEN Patterns	C Run-Time I	nterface C RAF	PR Global Comma	ands	
Keep Cancel Close ** ARL:RAPR Script ** ** Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands ** PERIODIC PERIODIC(vrate> <size)< td=""> rate zvoojy size gurz ** POISSON POISSON(vareRate (msg/sec) < size (bytes)>) aveRate (msg/sec) 1024 size (bytes) 4094 ** BURST BURST [REGULAR]RANDOM <aveinterval (sec)=""> (patternType) [<patternparams]< td=""> aveInterval (sec) 10 patternParams [cryegxdn] ** FXEDIEXPONENTIAL <aveduration (sec)="">) aveDuration (sec) 10 patternParams [cryegxdn] ** FOISSON[1024 4096] Keep Cancel Close</aveduration></patternparams]<></aveinterval></size)<>								,
** ARL::RAPR Script ** Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands PERIODIC PERIODIC[(rate> <size)]< li=""> rate zvoojy size gurz POISSON POISSON(<averate (bytes)<="" (msg="" <="" <size="" li="" sec)=""> aveRate (msg/sec) 1024 size (bytes) patternParams gurz patternParams gurz patternParams gurz patternParams gurz patternParams gurz g</averate></size)]<>						Keep Cancel	Close	
* ARL::RAPR Script × Reception Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands PERIODIC PERIODIC((rate> <size))< li=""> rate zvoojy size gurz POISSON POISSON(<averate (bytes)-)<="" (msg="" <size="" li="" sec)=""> aveRate (msg/sec) gurz <l< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></l<></averate></size))<>								
ARL::RAPR Script Reception Events Behavior Events Behavior Events RAPR Events MGEN Patterns Run-Time Interface RAPR Global Commands rate zvoojy size quuz rate zvoojy size (bytes) aveRate (msg/sec) 1024 size (bytes) patternParams cryegxdn FixED[EXPONENTIAL <aveduration (sec)=""> aveDuration (sec) To Keep Cancel Close</aveduration>								
C Reception Events C BAPR Events MGEN Patterns C Run-Time Interface C RAPR Global Commands C PERIODIC FERIODIC FERIODIC(state> <size)< td=""> rate zvoojy size gunz C POISSON POISSON(<averate (bytes)="" (msg="" <size="" sec)="">] aveRate (msg/sec) 1024 size (bytes) 4094 patternParams patternParams cryegxdn FNXEDIEX/PONENTIAL size(bytes) aveDuration (sec) 10 patternParams Cryegxdn F0ISSON[1024 4096] Keep Cancel Close</averate></size)<>	7/ ARL::RAPR Script							×
C PERIODIC PERIODIC (rate> <size>] rate zvoojv size qvurz i PDISSDN POISSDN(<averate (bytes)="" (msg="" <size=""]<="" sec)="" td=""> aveRate (msg/sec) 1024 size (bytes) 4094 C BURST BURST(REGULAR RANDOM <aveinterval (sec)="" <="" patterntype=""> [<patternparams>] aveInterval (sec) 10 patternParams cryegxdn FXEDIEXPONENTIAL <aveduration (sec)="">] aveDuration (sec) 10 patternParams cryegxdn POISSON[1024 4096] Keep Cancel Close</aveduration></patternparams></aveinterval></averate></size>	C Reception Events C Behavior Eve	ents 🔿 RAPF	l Events 💿 MGE	N Patterns	C Run-Time Inte	erface (C RAPR Global	Commands
• POISSON POISSON(size (bytes) aveRate (msg/sec) 4094 • BURST BURST[REGULARIRANDOM size (bytes) patternType (jozi patternType (jozi patternType (jozi patternParams)) aveDuration (sec) 10 patternParams (cyegxdn patternParams) (cyegxdn patternType (jozi patternParams)) aveDuration (sec) 10 POISSON[1024 4096] FixEDiExPONENTIAL <aveduration (sec)=""> (patternParams) (sec) = (patter</aveduration>	PERIODIC PERIODIC[<rate> <size>]</size></rate>			rate	zvoojy	size qyurz		
C BURST BURST[REGULARIRANDOM <aveinterval (sec)=""> <pre>cpatternType> [<ppre>cpatternParams>] aveInterval (sec) 10 patternType [qiozi patternParams cryegxdn FIXED[EXPONENTIAL <aveduration (sec)="">] POISSON[1024 4096] Keep Cancel Close</aveduration></ppre></pre></aveinterval>	POISSON POISSON[<averate (msg="" sec)=""> <siz< td=""><td>ze (bytes)>]</td><td>av</td><td>eRate (msg/sec)</td><td>1024 size (</td><td>bytes) 4096</td><td>-</td><td></td></siz<></averate>	ze (bytes)>]	av	eRate (msg/sec)	1024 size (bytes) 4096	-	
FIXED[EXPONENTIAL <aveduration (sec)="">] aveDuration (sec)]10 POISSON[1024 4096] Keep Cancel Close</aveduration>	BURST BURST[REGULAR RANDOM <avel< td=""><td>nterval (sec)> <patternt< td=""><td>ype> [<patternparams>]</patternparams></td><td>aveInterval (sec)</td><td>10 patterr</td><td>nType gjozi</td><td>patternParan</td><td>ns cryegxdn</td></patternt<></td></avel<>	nterval (sec)> <patternt< td=""><td>ype> [<patternparams>]</patternparams></td><td>aveInterval (sec)</td><td>10 patterr</td><td>nType gjozi</td><td>patternParan</td><td>ns cryegxdn</td></patternt<>	ype> [<patternparams>]</patternparams>	aveInterval (sec)	10 patterr	nType gjozi	patternParan	ns cryegxdn
POISSON[1024 4096] Keep Cancel Close	FIXED[EXPONENTIAL <aveduration< td=""><td>n (sec)>]</td><td>a</td><td>veDuration (sec)</td><td>10</td><td></td><td></td><td></td></aveduration<>	n (sec)>]	a	veDuration (sec)	10			
Keep Cancel Close	POISSON[1024 4096]							
						Кеер	Cancel	Close
MGEN Patterns			MGEN Patterns					

Figure 5. Editing a RAPR script by selecting an MGEN pattern.

% ARL::RAPR Script				×	
Reception Events O Behav	vior Events 💿 🔿 RAPR Even	nts 🔹 O MGEN Patterns	O Run-Time Interface O RAPR 6	Global Commands	
O JOIN <	<eventtime></eventtime>	eventTime	37.41		
[STOP <stoptime> DURATIO</stoptime>	N <duration>] stopTime</duration>	33.89 duration scyhyx		
d and a second se	JOIN <groupaddress></groupaddress>	groupAddress	sjomw		
[[INTERFACE <interfacename>]</interfacename>] interfaceName	kzycsb		
C LISTER .	[PORT <portnumber>]</portnumber>	portNumber	srypai		
U LISTEN «	(eventiime) (CTOD (starting) (DUD)(TIO)	event i me	88.32		
	(STUP < stop time> (DURATIO) (ISTEN < protocol> < portlist>	N < duration>j stop i me protocol	4.43 duration nzxrcjks		
		protocor	ponest group		
	74 ARL::RAPR Script				×
	C Reception Events	O Behavior Events 🛛 🔿 F	APR Events C MGEN Patterns	C Run-Time Interface 🛛 💿 R	APR Global Commands
<u> </u>	0	OVERWRITE_RAPRLOG	OVERWRITE_RAPRLOG <logfile></logfile>	logFile ysxmni	
	0	HOSTID	HOSTID <hostid></hostid>	hostid mwyzumu	
	C	RXBUFFER	RXBUFFER <rxbuffersize></rxbuffersize>	rxBufferSize bdltkvscr	
	C	MGENLOG	MGENLOG <logfile></logfile>	logFile Ivjgvyio	
	0	TOS	TOS <value></value>	value fmzwg	
	0	RAPRLOG	RAPRLOG <logfile></logfile>	logFile vkovk	
	0	TTL	TTL <value></value>	value kyudvylikp	
	0	LOAD_DICTIONARY	LOAD_DICTIONARY <dictionary-file.xml></dictionary-file.xml>	dictionary-file.xml 52.47	
	0	LABEL	LABEL <value></value>	value olaiewh	
	C	START	START <hour:min:sec>[GMT]</hour:min:sec>	hour:min:sec 11:04:21	
	0	OVERWRITE_MGENLOG	OVERWRITE_MGENLOG <logfile></logfile>	logFile nfdoxeh	
	0	OFFSET	OFFSET <seconds></seconds>	seconds 25	
	0	INTERFACE	INTERFACE <interfacename></interfacename>	interfaceName ikihdpdge	
	0	INPUT	INPUT <scriptfile></scriptfile>	scriptFile iokvgvlkjr	
	C	TXBUFFER	TXBUFFER <txbuffersize></txbuffersize>	txBufferSize cgfvljfyp	
					Keep Cancel Close
			RAPR Global Commands		

Figure 6. List of reception events and global commands.

74 ARL::RAPR Script	7% ARL::RAPR Script					
C Reception Ev	ents 💿 Behavior Events 💿 RAPR Ev	rents C MGEN Patterns	C Run-Time Interfa	ce O F	RAPR Global Commands	
C INTERROGATIVE	[<eventtime>]</eventtime>	eventTime	61.91			
	[STOP <stop_time> DURATION <duration>]</duration></stop_time>	stop_time	17.21 dur	ation ewszonv		
	INTERROGATIVE <protocol></protocol>	protocol	ijagjumb			
	RETRYINTERVAL <retry_interval> NUMRETRIES <#_retry_interval> NUMRE</retry_interval>	etries> retry_interval	pkmphgvnlz #_re	etries 21		
	SRC <srcport></srcport>	srcPort	72			
	DST <dstip>/<dstport> <mgenpattern></mgenpattern></dstport></dstip>	dstiP	44 ds	tPort 57	mgenPattern jojcrzw	
	[SUCCESS <success_logic_id>]</success_logic_id>	success_logic_id	tbghqqo			
	[FAILURE <failure_logic_id>]</failure_logic_id>	failure_logic_id	bwkxf			
	[PAYLOAD <payload_logic_id>]</payload_logic_id>	payload_logic_id	hetrnjdtro			
	[TIMEOUT <timeout_logic_id>]</timeout_logic_id>	timeout_logic_id	29.59		_	
	[TTL <timetolivevalue>][TOS <tos>]</tos></timetolivevalue>	timetolivevalue	47.28	tos eszdi		
C DECLARATIVE	<eventtime></eventtime>	eventTime	18.23			
	[STOP <stop_time> DURATION <duration>]</duration></stop_time>	stop_time	24.60 dur	ation dpgvjenh		
	[RAPRFLOWID <raprflowid>]</raprflowid>	raprFlowId	yaedoqlouj		_	
	DECLARATIVE <protocol> SRC <srcport></srcport></protocol>	protocol	jhbwkovp sro	Port 97		
	DST <dstip>/<dstport> <mgenpattern></mgenpattern></dstport></dstip>	dstiP	12 ds	tPort 86	mgenPattern wixxkhspfj	
	[SUCCESS <success_logic_id>]</success_logic_id>	success_logic_id	niadwtkr			
	[FAILURE <failure_logic_id>]</failure_logic_id>	failure_logic_id	hltmoz			
	[PAYLOAD <payload_logic_id>]</payload_logic_id>	payload_logic_id	rssqpxmbqx			
	[TIMEOUT <timeout_logic_id>]</timeout_logic_id>	timeout_logic_id	30.71		_	
	[TTL <timetolivevalue>][TOS <tos>]</tos></timetolivevalue>	timetolivevalue	67.87	tos bjhrm		
C STREAM	[<eventtime>]</eventtime>	eventTime	84.42			
	[DURATION <duration>]</duration>	duration	wfgvaiw		_	
	STREAM RESPPROB <lowrange> <highrange></highrange></lowrange>	lowRange	iaxveo highRa	ange nwdjxjd		
	[BURSTPRIORITY < burstPriority>]	burstPriority	wyswruvr			
	[BURSTDURATION <burstduration>]</burstduration>	burstDuration	qjowkjrr			
	[BURSTCOUNT <burstcount>]</burstcount>	burstCount	xtgebx		_	
	[BURSTDELAY <lowdelay> <highdelay>]</highdelay></lowdelay>	lowDelay	cavfht highD	elay xhiggleb	_	
	[BURSTRANGE <lowrange> <highrange>]</highrange></lowrange>	lowRange	jiaxveo highRa	ange [nwdjxjd		
	[TIMEUUTINTERVAL <timeoutinterval>]</timeoutinterval>	timeoutInterval	80.98	P P	_	
	<pre><pre>cprotocol> SRU <srcport></srcport></pre></pre>	protocol	msnkvipi sro	Port 93		
	DST <dstip>/<dstport> <mgenpattern></mgenpattern></dstport></dstip>	dstiP	34 ds	tPort [11	mgenPattern [ironfph	
C. 05010010	[PAYLUAD <payloadlogicid>]</payloadlogicid>	payloadLogicid	zgmvyitiz			
O PERIODIC	<event1ime></event1ime>	eventTime	32.63		_	
	[STUP <stop_time> [DURATION <duration>]</duration></stop_time>	stop_time	21.15 dur	ation gjyvglo		
	[HAPHFLUWID <raprhowld>]</raprhowld>	raprHowld	qtznch			
	PERIUDIC IN LERVAL (Interval)	interval	uxtzz			
	DURATION (duration)	duration	givvgio			
		Deciarative/Interrogative Description	Dypevakxiu	1		
				Кеер	Cancel Close	
		Behavior Events				

Figure 7. List of RAPR behavior events.

ARL::RAPR Script					
C Reception Events	C Behavior Events	RAPR Events	C MGEN Patterns	C Run-Time Interface	C RAPR Global Commands
RAPRFLOWID	<eventtime> RAPRFLOWID <raprflowid></raprflowid></eventtime>		eventTime 50.53	raprFlowId	sdnhosm
LOGICTABLE_FILE	<eventtime> LOGICTABLE_FILE <filename></filename></eventtime>		eventTime 63.64	fileName	nvuir
CHANGE_UBI_STATE	<eventtime> CHANGE_UBI_STATE <ubistate< td=""><td>></td><td>eventTime 93.79</td><td>ubiState</td><td>mzgwmer</td></ubistate<></eventtime>	>	eventTime 93.79	ubiState	mzgwmer
CLEAR	[<eventtime>] CLEAR <[ALL BehaviorEventT</eventtime>	ype RAPREVENT]>	eventTime 88.72	[ALL BehaviorEventType RAPREVENT]	ydspfd
STOP	STOP <hour.min:sec></hour.min:sec>		hour:min:sec 03:59:01		
LOGICID	<eventtime> LOGICID <logicid></logicid></eventtime>		eventTime 46.71	logicid	ppsmpc
RAPRPIPE	RAPRPIPE <pipename></pipename>		pipename ivofesqxx		
DICTIONARY_FILE	<eventtime> DICTIONARY_FILE <filename></filename></eventtime>		eventTime 84.67	fileName	xyhibthg
DICTIONARY_ENTRY	<eventtime> DICTIONARY_ENTRY <dictionar< td=""><td>yName> [<dictionaryvalue>]</dictionaryvalue></td><td>eventTime 90.30</td><td>dictionaryName</td><td>hlbacztdi dictionaryValue hgitwi</td></dictionar<></eventtime>	yName> [<dictionaryvalue>]</dictionaryvalue>	eventTime 90.30	dictionaryName	hlbacztdi dictionaryValue hgitwi
INPUT	<eventtime> INPUT <filename></filename></eventtime>		eventTime 9.46	fileName	ygziqyol
CHANGE_STATE	<eventtime> CHANGE_STATE <state></state></eventtime>		eventTime 15.55	state	tofglys
C PERIODIC PERIO C POISSON POISS C BURST BURS Fixed	DDIC[<rate> <size>] SON[<averate (msg="" sec)=""> <size (bytes)="">] T[REGULAR RANDOM <aveinterval (sec)<br="">JEXPONENTIAL <aveduration (sec)="">]</aveduration></aveinterval></size></averate></size></rate>	> <patterntype> [<pattern< th=""><th>aveRate (msg iParams>] aveInterval aveDuration</th><th>rate zvoojy size qyurz /sec) 0 size (bytes) qoyqt (sec) 10 patternType qiozi (sec) 10</th><th>jiw patternParams cryegxdn</th></pattern<></patterntype>	aveRate (msg iParams>] aveInterval aveDuration	rate zvoojy size qyurz /sec) 0 size (bytes) qoyqt (sec) 10 patternType qiozi (sec) 10	jiw patternParams cryegxdn
	L:RAPR Script			Кеер	Cancel Close
	Reception Events C Behavior Events	C RAPR Events	C MGEN Patterns	Run-Time Interface	Global Commands
	C START_RAPR_IN C SUBSEQUENT_I	NSTANCE RAPR inst NVOCATION RAPR inst	ance <raprinstanceid> ance <rapr1 event=""></rapr1></raprinstanceid>	RAPRinstanceID dwwrkygs RAPR1 event 21.06	eep Cancel Close
		Run	Time Interface		

Figure 8. List of RAPR events, MGEN patterns, and run-time interfaces.

3. Conclusion

The successful development of the ARL RaprEdt tool facilitates the creation RAPR files at ARL. Using the tool, a communication engineer can focus on the development of a communication scenario without being concerned with the mechanical structure of the files; especially, the XML files. Once a dictionary and a logic table are created, their contents are embedded in appropriate script files for specifying and running various communication scenarios.

Acronyms

ARL	U.S. Army Research Laboratory
FY08	fiscal year 2008
GUIs	graphical user interfaces
IA	information assurance
MANET	mobile ad hoc network
MGEN	Multi-Generator
NRL	U.S. Naval Research Laboratory
RAPR	Real-time Application Representative
VMs	virtual machines
WEL	wireless emulation laboratory
XML	extensible markup language

<u>No. of</u> Copies Organization

1 (PDF ADMNSTR

ONLY) DEFNS TECHL INFO CTR ATTN DTIC OCP (ELECTRONIC COPY) 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218

- 1 HC US ARMY RSRCH LAB ATTN AMSRD ARL CI OK TP TECHL LIB T LANDFRIED BLDG 4600 APG MD 21005-5066
- 6 HCs US ARMY RSRCH LAB ATTN AMSRD ARL CI OK T TECHL PUB ATTN AMSRD ARL CI OK TL TECHL LIB ATTN IMNE ALC IMS MAIL & RECORDS MGMT AMSRL ARL CI NT N IVANIC R HARDY B NGUYEN ADELPHI MD 20783-1197

Total: 8 (1 PDF, 7 HCs)