1

BBN Systems and Technologies
A Division of Bolt Beranek and Newman Inc.

BBN Report No. 7352
Appendix C

SIMNET CVCC

SIMNET SIMULATION OF RADIO COMMUNICATION: A TESTBED FOR INVESTIGATION OF C3I TECHNOLOGY

APPENDIX C: RADIO PERFORMANCE MONITOR

bbn

Report No. 7352

SIMNET CVCC

SIMNET SIMULATION OF RADIO COMMUNICATION: A TESTBED FOR INVESTIGATION OF C3I TECHNOLOGY

APPENDIX C: RADIO PERFORMANCE MONITOR

July 1991

Prepared by:

BBN Systems and Technologies
Advanced Simulation
10 Moulton Street
Cambridge, MA 02138 USA

Prepared for:

Defense Advanced Research Projects Agency (DARPA)
1400 Wilson Blvd.
Arlington, VA 22209-2308

This research was performed by BBN Systems and Technologies under Contract Nos. MDA972-89-C-0060 and MDA972-90-C-0061 to the Defense Advanced Research Projects Agency (DARPA). The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policy, either expressed or implied, of DARPA, the U.S. Army, or the U.S. Government.

Table of Contents

List of Figures

C.1 Introduction

The radio performance monitor (**radmon**) is a program for monitoring and analyzing the performance of the SINCGARS radio simulation for SIMNET. **Radmon** monitors traffic on the simulation network and records information broadcast by the SINCGARS radio simulator hosts concerning every important state change of every simulated radio. For example, when the channel selection knob on a radio is turned to a different channel, the SINCGARS radio simulator broadcasts a TransmitterPDU giving the new frequency selection for that radio. **Radmon** receives the transmission and records which radio changed state, the nature of the change, and the time at which the change occurred. **Radmon** maintains a running history of these state changes for each radio.

The information that **radmon** records can be saved in a file and subsequently restored from that file. **Radmon** allows the user to browse through the recorded data to examine interactions in detail. The recorded data can also be summarized to show channel utilization, radio utilization, and other statistics of radio activity.

The collected information can be presented in several ways. A stripchart display shows the activity of radios as a function of time. A connectivity map shows the the communication paths that are usable. A status display shows the detailed state of individual radios.

C.2 Summary of SINCGARS Radio Simulation

The simulation of SINCGARS radios is performed by one or more radio simulation hosts. Details of this simulation are given elsewhere, but those aspects which are pertinent to **radmon** will be summarized here. Each radio simulation host simulates the operation of one or more SINCGARS radios. The simulation host receives digitized voice data from microphones in the simulated vehicle or standalone radio and broadcasts this data whenever the push-to-talk button the microphone is pressed. When the simulation hosts receive this data, they play it through the earphones or speakers connected to the those radios which the simulation indicates are capable of receiving the signal.

For each simulated radio, the simulation host monitors the position of various knobs and buttons on a front panel. These knobs and buttons allow the selection of radio channels, frequencies, power levels, and so on. As the user changes the settings of the knobs or pushes the buttons, the simulation host notices these changes, incorporates the changes into the state information it maintains and broadcasts a "transmitter PDU" giving the new state so that the other radio simulation hosts also know the state of every simulated radio in the network.

The radio simulation hosts also monitor the "vehicle appearance PDUs" broadcast by the vehicle simulators to determine the position of the vehicles and hence the radios in those vehicles. The radio simulation hosts also broadcast vehicle appearance PDUs for standalone radios which are not installed in vehicles. The position information contained in the vehicle appearance PDUs allows the signal attenuation from a transmitter to a potential receiver to be computed using a radio propagation model. The attenuation and transmitter power level allows the received signal strength to be determined for every active transmitter. From the received signal strength, the radio tuning, and the assumed receiver sensitivity, the radio simulation host can determine which signal, if any a given radio can receive.

The radio simulation host broadcasts a "receiver PDU" each time the signal a radio is receiving changes. The receiver PDU specifies which transmitter is being received and its signal strength. In the case that all signals are too weak to be received, a receiver PDU is broadcast specifying which transmitter has strongest signal and what the received signal strength is. These receiver PDUs are not used by the radio simulation, but are broadcast specifically to provide information for the use of data loggers and programs such as **radmon.**

C.3 Data Acquisition

Radmon monitors the transmitter and receiver PDUs broadcast by the radio simulation hosts and the vehicle appearance PDUs broadcast by the vehicle and radio simulation hosts. As each PDU is received, the state of the affected radio or radios is changed. A radio's state consists of the following information:

- Tuning (frequency or hopset/lockout set).
- Status. One of the following:

non–existant	No TransmitterPDU received for 12 seconds.
inactive	Existant and operative but not doing anything.
inoperative	The vehicle or its antenna has been destroyed.
receiving	The radio is receiving a tranmission.
not–receiving	The radio is not receiving a transmission because the received power is less than the noise power. Receiver noise (sensitivity) and interfering transmissions all contribute to the noise power. The strongest transmitter is identified as the transmitter which is not being received.
transmitting	The radio is transmitting.

- Speaker (who's talking if the status is transmitting).
- Antenna height.
- Transmitted power.
- Received power (if receiving or not–receiving).
- Who is transmitting (if receiving or not–receiving).
- Geographic position.

The new state is found in or entered into a hash table and the pointer to that hash table entry along with the time at which the PDU was received is appended to the timeline for that radio. A separate timeline is maintained for each radio. As new radios appear in the network, a new timeline is started for that radio. The current implementation maintains all timelines in main memory. The timelines comprise the raw data from which radmon generates various displays.

C.4 Program Organization

Radmon is an X-windows application using the *Motif* toolkit. The content and layout of the user interface is specified using *Motif UIL* (User Interface Language). An X-windows resource file specifies details of the interface such as the words used to label the buttons and the text of headings can be determined at runtime. This accomodates personal preferences as well as allowing foreign language interfaces..

Radmon is a typical X-windows application in that it is event-driven. Various events such as pressing a mouse button or typing a key on the keyboard generates an event. The program dispatches on the event to perform an action appropriate to the event.

In addition to X-windows events, **radmon** uses the UNIX alarm clock signal to periodically wake up and process PDUs received from the simulation network. This interrupt occurs frequently enough to avoid data overruns from the network, but not so often that an excessive amount of overhead is incurred in such processing. The network interface hardware timestamps the incoming PDUs so that the delay before they are processed is inconsequential. The timestamps attached to the PDUs by the hardware are sufficiently accurate that the causality of simulation events is readily determined. For example, if a receiver stops receiving a transmission, **radmon** can determine if the cessation of reception is because the transmission stopped or because the signal was lost due to propagation effects. In the former case, **radmon** will receive a receiver PDU indicating the end of reception and a transmitter PDU indicating the end of transmission at about the same time. In the latter case, the end of transmission will not occur until much later.

As described above, state changes reported by the incoming PDUs are used to construct a timeline for each radio. These timelines constitute the raw data from which various displays are generated. We will describe the content of these displays and other aspects of the user interface to explicate the remaining aspects of the operation of the **radmon** program.

C.5 User Interface

The user interface to **radmon** consists of a window containing a panel of buttons on the left side and a scrollable area on the right. The panel of buttons is divided into three areas: At the top are the basic function buttons (see figure 1). In the middle are the sub–function buttons for those functions requiring sub–functions. For example, the States function has three sub–functions. The bottom contains a set of selection buttons which vary from function to function.

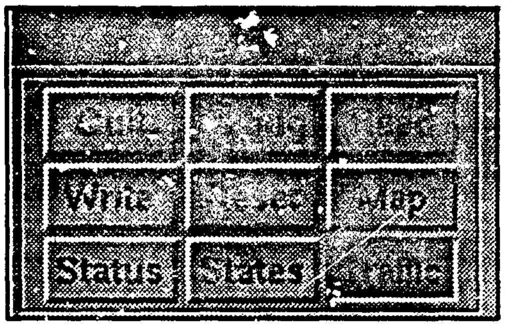


Figure 1: Radmon Function Buttons

The function buttons consist of the following:

Quit	Exits the program. Accumulated data is discarded.
Debug	Dumps debugging information useful to programmers.
Reset	Discards accumulated raw data and begins collecting new data.
Read	Reads raw data from a file. A file selection dialog is popped up and new raw data is read from the specified file. The old raw data is discarded. After reading a file, data arriving from the simulation network is ignored and only the data read from the file is available for display or analysis.
Write	Writes raw data to a file. A file selection dialog is popped up and the current raw data is written to the file. Data arriving from the simulation network while the file is being written may be lost, but additional data will be added to the raw data after the write operation is completed.
Map	A connectivity map is displayed. This is described in more detail below.
Status	Allows status descriptions for individual radios to be displayed. See below.
State	Generates a statistics summary. This is described in detail below.
Traffic	A traffic strip chart is displayed. This is described in detail below.

C.6 Connectivity Map Display

The connectivity map displays a map of vehicles with radios and their connectivity. This map does *not* represent the actual vehicle locations. In fact, usually the user must reposition the vehicles to avoid clutter on the screen. Each vehicle is represented by a rectangle on the screen. These rectangles are labelled with the vehicles' bumper numbers if available. If a bumper number is not available, that vehicle's rectangle is labelled with the vehicle id number. The background color of the vehicle rectangles is normally olive drab, but is changed to a grayish olive for vehicles which are apparently non-existant (no vehicle appearance PDU for more than 12 seconds). Figure 2 shows a typical connectivity map.

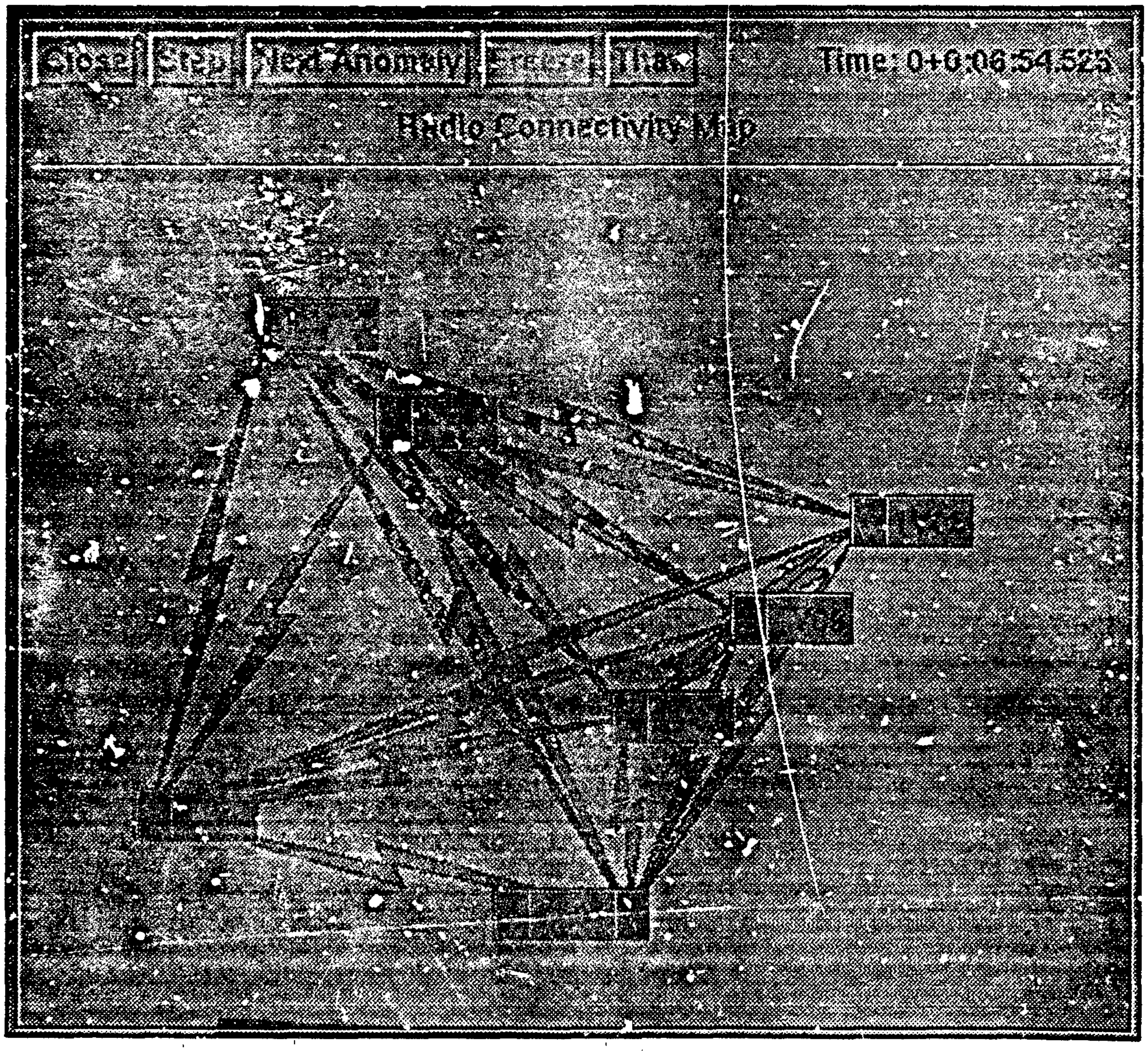


Figure 2: Radio Connectivity Map

Attached to either end of the vehicle rectangle is a rectangle for each radio the vehicle contains. The "A" radio is on the left side and the "B" radio is on the right. Vehicles with no "B" radio have no rectangle on the right side. Again, the background color of the rectangle denotes the existance of the radio. It is possible for a non-existant vehicle to have existant radios and vice-versa although this is anomalous and usually transient. A non-existant radio on an existant vehicle is usually due to the radio simulator for that radio being inoperative. A non-existant vehicle with an existant radio is also transitory since the radio simulator for that radio will note the missing vehicle appearance PDUs and cease transmitting transmitterPDUs for the radios of that vehicle.

The background color of the radio rectangle also denotes whether it is receiving, not-receiving, transmitting, or doing nothing. A transmitting radio is colored green, a receiving radio is colored magenta, a not-receiving radio is colored red, and a radio doing nothing is colored olive-drab.

The vehicle rectangles are initially placed on the connectivity map in a position corresponding to their relative positions in the world. That is a given vehicle will be positioned on the connectivity map to the left of all vehicles which are more westerly than itself and will be positioned above all vehicles which are more southerly than itself. However, the distance between vehicles on the map is not proportional to their actual separation. This is done because vehicles are often clustered into a few widely separated groups with the result that all the rectangles for vehicles in a group end up being displayed with nearly identical positions.

The vehicle rectangles may be moved by simply positioning the mouse over a vehicle rectangle, depressing the left button, dragging the rectangle to a new position, and releasing the button.

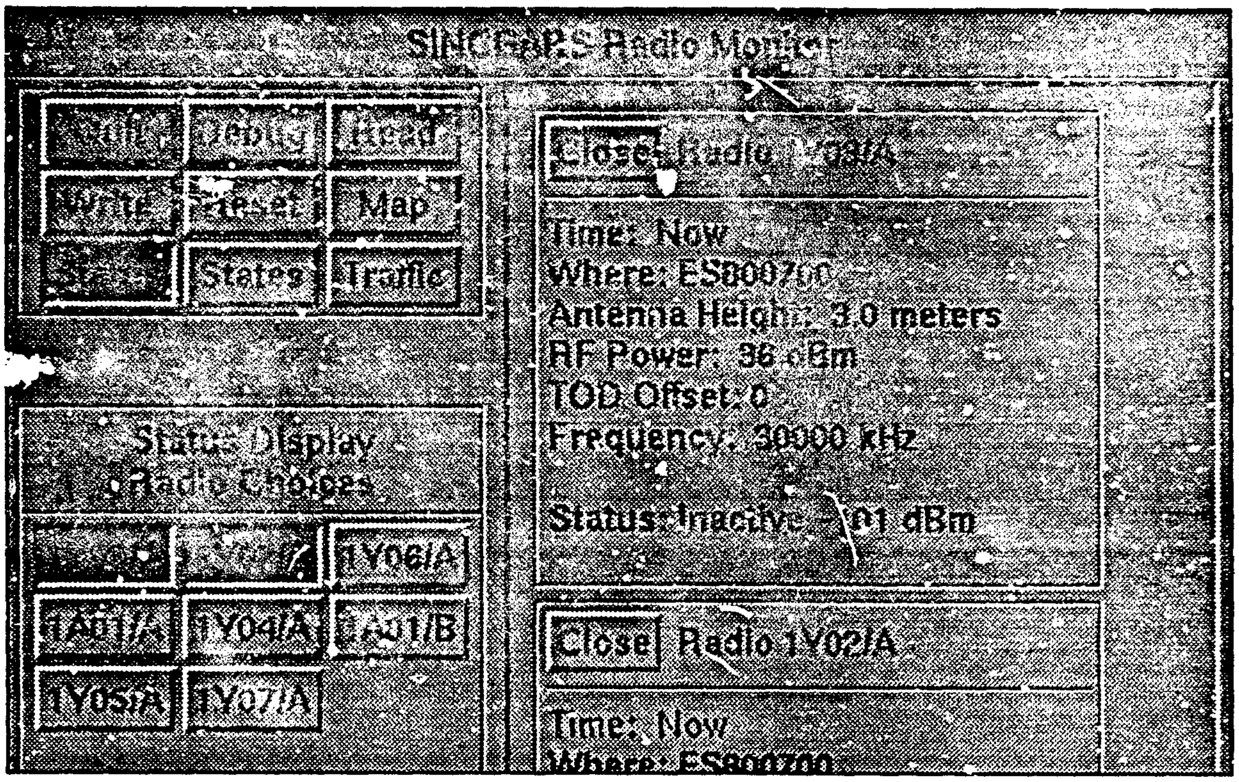
The state of connectivity between pairs of radios is represented by drawing a lightning bolt between the radios. The shape and color of this bolt represents the state of connectivity between the radios. Radios which are not tuned to the same frequency or hopset have nothing drawn between them. For radios which are tuned the same, but which have never transmitted to each other (while the radmon program has been running), the bolt is an empty outline. For radios which have successfully transmitting in one direction but never transmitted in the other direction, the bolt is green and pointed on only one end (the receiving end). For radios which have successfully transmitted both ways, the bolt is green and pointed on both ends. For radios which have failed to hear a transmitter, the bolt is colored red on the end which has failed to receive (both ends are colored red if neither has heard the other). For transmissions that are currently active, the bolt is striped green and black and animated in the direction of transmission. For transmissions that are currently active and for which reception has been preempted by another simultaneous transmission, the bolt is colored orange.

The map display is normally maintained in real time. As each radio changes state, the display is changed. In particular, at any moment the map display shows who is transmitting and receiving and, more importantly, who is not. During periods of rapid-fire communication, the display may lag behind. This happens for two reasons: First, the display update takes time and, second, we desire to display every state change for a minimum interval to insure its visibility. The lag in the display has a minimal effect on the acquisition of raw data.

The map display can also be frozen and stepped from one state change to the next or stepped to the next anomalous state. Anomalous states are those containing red or orange bolts. This stepping feature is best used in conjunction with the traffic display (see below). When the traffic display mark is moved and the connectivity map display is frozen, then the connectivity display shows the connectivity at the time selected in the traffic display. In this way the connectivity display can be used to review earlier anomalies.

C.7 Status Display

When the "status" button is selected, selection buttons for every radio are presented. Selecting these buttons, causes a box for each selected radio to be displayed (figure 3). This box describes the state of the radio and is updated whenever the state changes. Again, if the traffic display is frozen the status boxes display the state of the radios at the time selected by the traffic mark. Whenever the information in a box changes, the changed information is displayed in blue. Old information is shown in black.



Time: Now
Where: ES800700
Antenna Height: 3.0 meters
RF Power: 36 dBm
TOD Offset: 0
Frequency: 30000 kHz
Status: Inactive -101 dBm

Time: Now
Where: ES800700

Figure 3: Status Display

The selection buttons use a Motif selection policy known as extended selection. This means that if the mouse is positioned over a button and the left mouse button is clicked, that button becomes selected. Multiple buttons may be selected by dragging through several buttons. Normally, the previously selected buttons are cleared as each selection starts. This action is omitted if the control key is held down which the mouse button is pressed. In this case if the button where the selection starts was already selected, then the action deselects all the buttons which are dragged over. The most useful actions are to click on single buttons to select a radio for status display, drag from the first button to the last to select all radios, and to control-click on individual buttons to either select or deselect them.

C.8 State Display

The state display presents a summary of radio activity. When the "State" button is selected. A panel of three sub-function buttons appears. These buttons do the following:

radio	Presents radio selection buttons. Summary information is is presented for the radios selected from this panel. These are selection buttons; information about radios which are not selected is simply not presented.
attributes	Presents attribute selection buttons. The attributes selected by these buttons determine which states are included in the summary. These are filtering buttons; attributes which are not selected are ignored when distinguishing one state from another. The statistics for states which differ only in unselected attributes are combined.
status	Presents status type selection buttons. The selected status types are included in the summary. These are also a selection buttons; information about states which have a status which is not selected are simply not presented.

The state display consists of one or more rows of information. Each row summarizes one or more states of one or more radios. Each column contains the value of an attribute or statistic for those states. The statistics consist of the following:

%	The percentage of all time that is spent in this state. When radios are individually summarized (radio attribute selected) percentages are computed separately for each radio and thus the numbers in the column for one radio will sum to 100 (small rounding errors may occur). When the radios are aggregated (radio attribute not selected), the sum of all percentages will again sum to 100.
Count	The number of times a radio (or all radios) was (were) in this state.
Avg Time	The average duration of time spent in this state.
Max Time	The maximum duration of time spent in this state.

When an attribute is omitted all the states which are identical except for that attribute are combined. Statistics are combined as follows: *Count* is added. Total time in the combined state is coumputed by adding the times from the individual states and thus, % is added. Avg Time is computed from the combined total time and combined count. *Max Time* is the maximum duration of time spent in any of the combined states. In computing *Max Time*, the duration of time spent in a state is determined independent of which states might be combined. Therefore, a transition from one state to another with which it is being combined still counts as a state change; the durations are not added.

C.9 Traffic Display

The traffic display consists of a number of horizontal lines of data representing the state of radios as a function of time (see figure 4). Four states are distinguished: non–existant/inoperauve, inactive, transmitting, and receiving. The non–existant/inoperative/not–receiving state is is represented by a line with the same color as the background (which is therefore invisible). The inactive state is represented by a thin black line. The transmitting state is represented by a thick green line. The receiving state is represented by a thick magenta line.

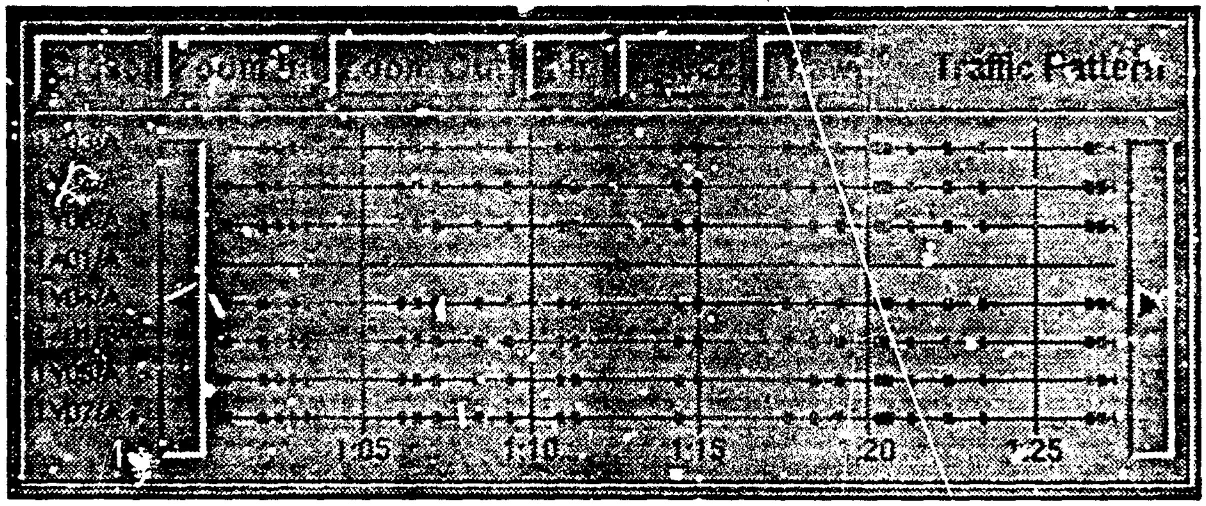


Figure 4: Traffic Display

When the traffic display is selected, two sub–function buttons appear:

Tuning Permits transmissions on particular frequencies or hopsets to be highlighted. Transmissions and receptions for unselected tunings are shown in black rather than color. Selecting no tuning buttons is equivalent to selecting them all.

Radio Selects which radios are included in the display.

The visible portion of the traffic display shows the data for some particular portion of time and normally scrolls to keep the present time at the right edge of the display. Arrows at either end of the strip allow the display to be scrolled left or right to expose a different part of the data. In addition, a scrollbar at the bottom of the display allows traversing the data by larger amounts: The scrollbar represents the entire available data and the slider represents the visible portion of the data. The slider may be dragged to the region of interest. Clicking the left button to the left of the slider moves the slider one page earlier in the data; clicking the left button to the right of the slider advances the slider by one page. Clicking the left button in the left arrow moves the slider a small amount earlier in the data and clicking the left button is the right arrow moves the slider a small amount later in the data. Depressing the shift key while clicking in either arrow moves the slider all the way to either the beginning of the data (left arrow) or the end of the data (right arrow). Buttons allow zooming in or out to change the scale factor of the display.

The traffic display allows a region of the display to be delimited using the mouse. If the left mouse button is pressed while positioned over the strip chart a mark iplaced at that point in time and shown as a blue vertical line over the chart. If, after the button is pressed, the mouse is moved left or right and then released, a second mark is made and again depicted as a blue vertical line. The "Zoom in" and "Zoom Out" buttons apply to the region thus delimited. "Zoom in" expands the image to make the delimited region fit between the left and right edges. "Zoom Out" shrinks the image to make the entire picture appear between the delimited extremes. When no region has been delimited, the zoom buttons zoom by a factor of two.

The "Fit" button causes the entire recorded data to be compressed into the display.

When the mark is positioned in the traffic display, the display is frozen and automatic scrolling ceases. The display will remain frozen until the "Thaw" button is selected. The display may also be frozen without setting the mark by selecting the "Freeze" button.

At very small scales, the duration of transmissions or receptions becomes very short. The traffic display is biased toward showing transmissions in preference to reception or inactivity. Therefore, at very small scales, the display will be mostly green.

C.10 Customization

Most of the factors controlling the operation of **radmon** are specified as X-windows resources and may be found in: /simnet/data/radmon/radmon.res. These values may be overridden using the -xrm option in the command invoking **radmon**. For example:

```
radmon -xrm '*exerciseID: 7'
```

Sets the exerciseID filter parameter so that **radmon** ignores all PDUs which are not part of exercise 7.

```
radmon -xrm 'radmon*status.hilitedColor: green'
```

Changes the color used to display new values in the status displays.

The following is a list of the resources for which values are specified in *radmon.res*. Those resources which are obvious or which are excessively complicated are not described. Also, all resource names begin with **radmon*** which has been omitted for brevity.

geometry: 800x600

The size and position of the **radmon** display window.

exerciseID: 255

The exercise ID of simulation PDUs which are monitored by **radmon**.

shadowThickness: 2

The thickness of the shadow around buttons.

ButtonLanimateInterval: 150

The interval (in milliseconds) between updates during scrolling of buttons.

radmon.onlineLabelFormat: SINCGARS Radio Monitor -- %v\nOnline Data\nMonitoring by %H (%E) on %d %M %y at %h:%m:%s

This is the format used for composing the main title. The following escape sequences are used:

%H	Host name — the name of the host on which **radmon** is running.
%E	The ethernet address on the simulation network.
%F	The name of the data file being perused or "online" if the data is being gathered online.
%v	The **radmon** version number.
%t	The time at which the data was acquired.
%w	The day of the week.
%M	The month.
%d	The day of the month.

%h	The hour.
%m	The minute.
%s	The second.
%Y	The four-digit year.
%y	The two-digit year.

radmon.offlineLabelFormat:	**SINCGARS Radio Monitor — %v\nData From File: %F\nMonitored by %H (%E) on %d %M %y at %h:%m:%s**

This is the format of the main title for data which comes from a file. The excape sequences are the same as for the online format.

reserveArrowSpace:	**False**

Specifies that the scrolling button lists should reserve a space for scrolling arrows even when they are not needed for a particular list.

font:	***helvetica-bold-r-normal--14*75*75***

The font used for text labels.

fontList:	***helvetica-bold-r-normal--14*75*75***

The fonts used for text labels. (No labels currently use more than one font).

ButtonL.borderWidth:	**0**

The width of the border around button labels.

function*buttons.numColumns:	**3**

The number of columns used for the function button list.

function*buttons.rows:	**3**

The number of rows used for the function button list.

function*buttons.hilitePolicy:	**HILITE_WHEN_SELECTED**

Causes buttons to be highlighted only when selected. Merely, passing the mouse over a button will not cause it to be highlighted.

function*buttons.quit:	**Quit**

The text of the label on the quit button.

function*buttons.dump:	**Debug**

The text of the label on the debug button.

function*buttons.read:	**Read**

The text of the label on the read button.

function*buttons.write:	**Write**

The text of the label on the write button.

function*buttons.reset:	**Reset**

The text of the label on the reset button.

function*buttons.map:	**Map**

The text of the label on the map button.

function*buttons.state: States

The text of the label on the states button.

function*buttons.traffic: Traffic

The text of the label on the traffic button.

function*buttons.status: Status

The text of the label on the status button.

function.width: 200

The width of the area for function buttons.

function.Spacing: 3

The spacing between function buttons.

state_attribute_functions*label1.labelString: State Display\nAttribute Choices

The label for the state attribute sub-function button list.

state_attribute_functions*selectionPolicy: EXTENDED_SELECT

The selection policy for the attribute selection buttons.

state_attribute_functions*hilitePolicy: HILITE_WHEN_SELECTED

See above.

state_status_functions*label1.labelString: State Display\nStatus Choices

The label for the state display status selection button list.

state_status_functions*selectionPolicy: EXTENDED_SELECT

state_status_functions*hilitePolicy: HILITE_WHEN_SELECTED

state_radio_functions*label1.labelString: State Display\nRadio Choices

state_radio_functions*selectionPolicy: EXTENDED_SELECT

state_radio_functions*hilitePolicy: HILITE_WHEN_SELECTED

status_functions*label1.labelString: Status Display\nRadio Choices

status_functions*selectionPolicy: EXTENDED_SELECT

status_functions*hilitePolicy: HILITE_WHEN_SELECTED

traffic_radio_functions*label.labelString: Traffic Display\nRadio Choices

traffic_radio_functions*selectionPolicy:	EXTENDED_SELECT
traffic_radio_functions*hilitePolicy:	HILITE_WHEN_SELECTED
traffic_tuning_functions*label.labelString:	Traffic Display\nTuning Choices
traffic_tuning_functions*selectionPolicy:	EXTENDED_SELECT
traffic_tuning_functions*hilitePolicy:	HILITE_WHEN_SELECTED
map_functions*label1.labelString:	Connectivity Display\nRadio Choices
map_functions*selectionPolicy:	EXTENDED_SELECT
map_functions*hilitePolicy:	HILITE_WHEN_SELECTED
map_functions*rows:	10
The number of rows of buttons in the selection lists.	
status.hilitedColor:	blue
The color of changed status values in status boxes.	
status.unhilitedColor:	black
The color of unchanged status values.	
status_rc.width:	300
The width of a status box.	
status_rc.shadowThickness:	0
The status_rc widget composes the status box from a heading, a separator and a list of labels. Change these resources at your peril.	
status_rc.entryBorder:	0
status_rc.borderWidth:	0
status_rc.adjustMargin:	False
status_rc.marginHeight:	0

status_rc.marginWidth: 0

status_rc.sepp.margin: 3

The space around the separator in the status display.

status_rc.heading.borderWidth: 0

The width of the border around the heading string.

status_rc.heading.close.labelString: Close

The label in the close button of a status box.

status_rc.body.marginWidth: 0

status_rc.body.marginHeight: 0

status_rc.body.spacing: –4

The spacing between lines in the status box. A negative spacing places the lines a little closer together than the default spacing. If this is too negative, ascenders or descenders may be lost. This may need to be adjusted if the font is changed.

status_rc.body.XmLabelGadget.labelString: From: Vehicle 101, Radio A––––

A prototype string used for sizing the labels in the status box.

status_rc.body.XmLabelGadget.marginHeight: 0

status_rc.body.XmLabelGadget.borderWidth: 0

status_rc.body.XmLabelGadget.recomputeSize: False

status_rc.body.XmLabel.labelString: From: Vehicle 101, Radio A––––

status_rc.body.XmLabel.marginHeight: 0

status_rc.body.XmLabel.borderWidth: 0

status_rc.body.XmLabel.recomputeSize: False

traffic.form.width: 485

The width of a traffic display.

traffic.form.Spacing: 5

The spacing between elements of the traffic display.

traffic.form.marginWidth:	**5**
The side margins for the elements of the traffic display.	
traffic.form.marginHeight:	**5**
The top and bottom margins for the elements of the traffic display.	
traffic.form.close.labelString:	**Close**
The label on the "close" button in a traffic display.	
traffic.form.zoom_in.labelString:	**Zoom In**
The label on the "zoom in" button.	
traffic.form.zoom_out.labelString:	**Zoom Out**
The label on the "zoom out" button.	
traffic.form.fit.labelString:	**\ Fit\**
The label on the "fit" button.	
traffic.form.freeze.labelString:	**Freeze**
The label on the "freeze" button.	
traffic.form.unfreeze.labelString:	**Thaw**
The label on the "thaw" button.	
traffic.font:	***-helvetica-medium-r-normal--10-*-75-75***
The font for labels in the traffic display.	
traffic.transmitBarColor:	**green**
The color of a traffic bar during transmission.	
traffic.transmitBarHeight:	**5**
traffic.receiveBarColor:	**magenta**
The color of a traffic bar during transmission.	
traffic.receiveBarHeight:	**5**
The height of a traffic bar during transmission.	
traffic.notReceiveBarColor:	**red**
The color of a traffic bar during non-reception.	
traffic.notReceiveBarHeight:	**0**
The height of a traffic bar during non-reception.	
traffic.untunedBarColor:	**dark slate grey**
The color of a traffic bar during transmission or reception not on one of the selected frequencies or hopsets.	
traffic.untunedBarHeight:	**5**
The height of a traffic bar during transmission or reception not on one of the selected frequencies or hopsets.	

traffic.silentBarColor: dark slate grey

The color of a traffic bar during silence.

traffic.cursorColor: medium blue

The color of the cursors in the traffic display.

traffic_sub_functions.inner.label.labelString: Traffic

The label on the traffic sub-function menu.

traffic_sub_functions.inner.List.hilitePolicy: HILITE_WHEN_SELECTED

The traffic sub-function button hilite policy.

traffic_sub_functions.inner.List.rows: 3

The number of rows of traffic sub-function buttons.

traffic_sub_functions.inner.List.numColumns: 2

The number of columns of traffic sub-function buttons.

state_sub_functions.inner.label.labelString: State

The label on the state sub-function menu.

state_sub_functions.inner.List.hilitePolicy: HILITE_WHEN_SELECTED

The state sub-function button hilite policy.

state_sub_functions.inner.List.rows: 3

The number of rows of state sub-function buttons.

sub_function_rc.*.inner.width: 188

The width of the sub-function button area.

sub_function_rc.*.inner.height: 90

The height of the sub-function button area.

state.form.Spacing: 5

The spacing between elements of the state display.

state.form.marginWidth: 5

The side margins for elements of the state display.

state.form.marginHeight: 5

The top and bottom margins for elements of the state display.

state.form.body.resizePolicy: RESIZE_NONE

The size of the state display is determined by its contents (rather than vice-versa).

state.form.label.labelString: Radio Statistics

The label on the state display

state.form.close.labelString: Close

The label on the "close" button of the state display.

state.font: *helvetica-medium-r-normal--12*75*75*

The font used in the state display.

state*body.width:	**500**

The default width of the state display. In practice, the width is determined from the contents.

state*body.marginWidth:	**0**

The side margins in the data portion of the state display.

state*body.marginHeight:	**0**

The top and bottom margins in the data portion of the state display.

map.vehicleColor:	**#809040**

The color of vehicle and inactive radio rectangles in the map display (dark olive drab).

map.manuallyPlacedVehicleColor:	**#90a000**

The color of manually placed vehicle and inactive radio rectangles in the map display (light olive drab).

map.nonExistantColor:	**#90a080**

The color of non–existant vehicle and inactive radio rectangles in the map display.

map.transmittingColor:	**green**

The color of transmitting radio rectangles in the map display.

map.receivingColor:	**magenta**

The color of receiving radio rectangles in the map display.

map.notReceivingColor:	**red**

The color of not–receiving radio rectangles in the map display.

map.unconnectedColor:	**red**

The color of bolts between unconnected radios.

map.connectedColor:	**green**

The color of bolts between connected radios.

map.conflictColor:	**orange**

The color of bolts between radios when a transmission conflict occurs.

map.receiving0Color:	**green**
map.receiving1Color:	**green**
map.receiving2Color:	**black**
map.receiving3Color:	**black**

One of the colors of bolts between radios during reception. The bolt is striped with a sequence of four colors. Animation is achieved by cycling the color map so each stripe is colored by each of the four colors in sequence.

map.untunedStyle:	**BLANK**

The style of the lightning bolts between radios not tuned the same.

map.tunedStyle:	**HOLLOW**

The style of the lightning bolts between radios tuned the same but for which no transmission have occurred..

map.unconnectedStyle:	**FILLED**

The style of the lightning bolts between radios that are not able to communicate.

map.connectedStyle: FILLED

The style of the lightning bolts between radios that are able to communicate.

map.conflictStyle: FILLED

The style of the lightning bolts between radios for which communication was interferred with by another simultaneous transmission.

map.receivingStyle: ANIMATE

The style of the lightning bolts between radios during reception.

map.foreground: black

The color of outlines and other foreground objects in the map display.

map.animationInterval: 100

The interval between color map changes used for animation.

map.form.Spacing: 5

The spacing between separators and other elements of the map display.

map.form.marginWidth: 5

The side margin of separators and other elements of the map display.

map.form.marginHeight: 5

The top and bottom margins of separators and other elements of the map display.

map.form.body.resizePolicy: RESIZE_NONE

The size of the map is constant.

map.form.label.labelString: Radio Connectivity Map

The text of the label on the map display.

map.form.close.labelString: Close

The text of the label on the "close" button.

map.form.step.labelString: Step

The text of the label on the "step" button.

map.form.anomaly.labelString: Next Anomaly

The text of the label on the "next anomaly" button.

map.form.freeze.labelString: Freeze

The text of the label on the "freeze" button.

map.form.unfreeze.labelString: Thaw

The text of the label on the "thaw" button.

map*form.width: 512

The width of the map display.

map*body.height: 384

The height of the map display.

map*body.marginWidth:	0
map*body.marginHeight:	0
map*body*form.marginHeight:	0
map*body*form.marginWidth:	0
map*body*form.spacing:	0
map*body*form.XmSeparator.borderWidth:	0
map*body*form.XmSeparatorGadget.borderWidth:	0
map*body*form.XmLabel.marginHeight:	0
map*body*form.XmLabelGadget.marginHeight:	0
map*body*form.XmLabel.marginWidth:	0
map*body*form.XmLabelGadget.marginWidth:	0
radmon.disp*file.dirMask:	*.radmon
The pattern for scanning a directory of radmon data files. Files matching this pattern a available for reading or overwriting.	
functions.inner.List.adjustLast:	false
general.scrolledWindowMarginWidth:	5
general.width:	650
The width of the general display area.	
general.height:	500
The height of the general display area.	
work.width:	620
The width of the work area within the general display area.	
work.height:	10

The height of the work area within tne general display area.

work.marginWidtn:	**10**
work.marginHeight:	**10**
work.spacing:	**5**
work.borderWidth:	**0**