Scriptools User's Guide

Brian Adamson, Naval Research Laboratory

1. Overview

This document describes a set of tools that were developed to support generation and manipulation of "planning" and "scripting" documents and files used in support mobile network modeling. The tool set supports mobile network modeling using emulation environments such as the Naval Research Laboratory (NRL) "extensible Mobile Ad-hoc Network Emulator" (eMANE) framework or using network simulation tools such as *ns-2* . An XML Schema document that is part of this distribution specifies some different XML document types that can be used to plan and script mobile network modeling scenarios. These "Scriptools" support generation, manipulation, and parsing of these XML document types. Additionally, some ancillary tools are provided to manipulate some other file types used in mobile network emulation and simulation such as the NRL "Scripted Display Tool" (SDT) format and *ns-2* TCL mobility scripting files. In some cases, these tools support conversions among some different formats. Additional formats will be supported as these tools evolve.

In addition to file manipulation, it is also planned that options will be added to the toolset described here to also provide run-time generation of "Event" messaging that is used to orchestrate and control eMANE experiments. Similarly, tools may be eventually provided to perform monitoring of selected eMANE run-time "events" and perhaps generate other events or perform data logging or other actions in response.

The initial focus of this tool set is on planning, scripting, and generating mobile node location and motion properties for wireless network emulations and simualtions. However, some rudimentary capability to generate connectivity information based on radio propagation "path loss" computation or other criteria (e.g. "comms range") is also provided.

As mentioned above, an XML Schema document (`EmulationScript.xsd`) also accompanies the tool distribution as well as a document (`EmulationScriptSchema.pdf`) that provides a detailed description of the document types specified and their intended use. Please refer to that document as reference material for those document types. As noted, the principal purpose of some of the tools described here is related to manipulation of those document types.

2. MotionPlanner (*mp*) Tool

The MotionPlanner (*mp*) tool parses the *MotionPlan* XML document type and outputs a corresponding *EmulationScript* XML document containing the time-ordered set of "`EmulationScript:Events`" that script mobile node location and motion. The *MotionPlan* document contains loose descriptions of intended mobile node location and motion on a "nodal basis" while the *EmulationScript* provides stricter, time-ordered specification of mobile node location and motion property updates (as well as potentially other "Node" or emulation module property updates).

2.1. Usage

The usage of the *mp* tool is summarized by:

```
mp [vector] input <inFile> [input <inFile1> input <inFile2> ...]
```

The required <inFile> parameter must be a valid *MotionPlan* document. Multiple <inFile> *MotionPlan* documents may be processed by using the "input" directive multiple times on the command line. The input files are processed and a single *EmulationScript* document is generated that represents the union of the *MotionPlan* set.

The optional "vector" key word indicates that MotionPlan:vector primitives should be output as EmulationScript:vector motion types. The default behavior (when "vector" is not specified) is that *mp* translates MotionPlan:vector motion primitives into the EmulationScript:waypoint motion type. The purpose of this is to better serve the *EmulationScript* philosophy of providing complete and deterministic (as possible) information for easier run-time manipulation when *EmulationScript* documents are used to drive run-time emulation (or simulation) execution. The "vector" option provides a way to override this default behavior.

Table 1. mp command-line options

input <inFile>	Specifies an input filename <inFile> to be processed. Note the "input" directive can be given multiple times on the command line to merge multiple *MotionPlan* documents into a single *EmulationScript* output. The input file(s) must be valid *MotionPlan* XML document(s).
vector	When specified, MotionPlan:vector motion primitives in the input file are converted to the corresponding EmulationScript:vector motion type in the output file. When not specified (the default), MotionPlan:vector motion primitives are converted to equivalent EmulationScript:waypoint motion type instances. The EmulationScript:waypoint motion type provides a more explicit description of the node's location over time that is expected to be preferrable for many uses of *EmulationScript* documents.

The resultant *EmulationScript* is directed to the program's STDOUT. For example:

```
mp input motionPlan.xml > emuScript.xml
```

will generate an *EmulationScript* document named "emuScript.xml" from the "motionPlan.xml" file.

An option to take an existing *EmulationScript* document and add additional EmulationScript:Events to that document from one or more *MotionPlans* will be provided in the future. Similarly additional "planning" document types (e.g. *CommunicationPlan*, *ExperimentPlan*, etc) may be defined in the future and a similar, iterative process of generating an *EmulationScript* from a superset of *MotionPlan* and other such "planning" documents will be supported. Thus the *mp* tool may become one of a family of plan->script generation tools. Related to this, a mechanism is being developed to allow inter-dependencies among planning documents to be expressed and managed.

3. MotionGenerator (*mg*) Tool

The MotionGenerator (*mg*) tool parses the *EmulationScript* XML document type and outputs "node location" events at a specified time interval. The currently support output formats are the SDT format and a skeleton *ns-2* TCL script for specifying mobile node motion using the *ns-2* "setdest" command. Note that SDT format output consists of a series "node position" commands paced by the SDT "wait <msec>" command to convey real-time motion. Future revisions of this tool may also provide options to output other file formats and/or generate run-time eMANE "location events".

3.1. Usage

The usage of the *mg* tool is summarized by:

```
mg input <fileName> [interval <updateInterval>][duration <maxTime>][format {sdt|ns2}]
```

The required `<inFile>` parameter must be a valid *EmulationScript* document ...

Table 2. mg command-line options

`input <fileName>`	Specifies the input file `<fileName>` to be processed. The file must be a valid *EmulationScript* XML document.
`interval <updateInterval>`	This specifies the update interval (in seconds) used in generating mobile node location updates to the output. The default is node location is generated once per second. Note that node location updates are generated only when a node has actually moved (or for its intial location).
`duration <maxTime>`	This specifies the maximum time (in seconds) for which node locations are generated. A `<maxTime>` value of -1.0 indicates that location updates are generated until the entire script is executed. This is the default behavior when a "`duration`" is not specified.
`format {sdt | ns2 | simple}`	This sets the desired output format that is directed to `STDOUT`. The SDT format is default.

Three different output formats are supported:

1. The "sdt" output puts out a mobility trace using the SDT file format with SDT "node position" and "wait" commands to convey the position of the nodes and timing of their location changes at the `updateInterval`.

2. The "ns2" format outputs a TCL script using *ns-2* "`setdest`" and related commands. Note that for "ns2", nodes named in a convention of "node01", "node02", "n9", etc will be respectively identified as dereferenced TCL variables in the form "$node(1)", "$node(2)", "$n(9)", etc.

3. Finally, the "simple" format is a text format with lines in the form "`time>seconds node>nodeName location>x,y[,z]`" where "seconds" is a floating point value of the time in seconds, "nodeName" is the node identifer from the XML script, and the "x,y[,z]" are the node coordinate values ("z" is optional).

The "mp" tool can be used with the example MotionPlan documents included in the distribution to generate the corresponding "EmulationScript" documents which, in turn, can be used with the "mg" tool to generate corresponding "sdt', "ns-2", or "simple" mobility traces.

4. GraphBuilder (*gb*) Tool

The GraphBuilder (*gb*) tool parses a pre-existing SDT or *ns-2* file that contains mobile node locations and generates an output file in either the SDT format (with added SDT "link" and "unlink" commands to indicate connectivity) or the MITRE Mobility Format (MMF) (which conveys nodes locations and their propagation path losses to other nodes over time). The name "GraphBuilder" is used since both of these file formats with the embedded node connectivity information essentially describe a possibly dynamic "graph". The SDT file format is a text file containing "node position", "link/unlink", and other commands as described in the SDT documentation at http://pf.itd.nrl.navy.mil/protools/sdt.html.

These tools and some other related NRL tools used the SDT format as a convenient file format for describing "graphs" of "nodes" (vertices) with dynamic connectivity (edges). Future versions of this tool may support additional "graph" formats as needed. The support for the MMF format is principally to support the initial release of the eMANE software suite that used this format as a run-time script to control connectivity among emulated mobile networking nodes.

4.1. Usage

The usage of the *gb* tool is summarized by:

```
gb input <mobilityTraceFile> [gps][range <commsRange>][mmf|sdt][speed <factor>]
```

The required <mobilityTraceFile> parameter must be a text file containing SDT commands or *ns-2* node motion directives (e.g. "setdest"). The output is an SDT file that mirrors the input with respect to node location but adds connectivity information with SDT "link" and "unlink" commands. Similarly, the MMF format which contains embedded path loss values among the mobile nodes provides yet more detailed connectivity information.

Table 3. gb command-line options

input <mobilityTraceFile>	Specifies the input filename <inFile> to be processed. SDT and ns-2 input files are currently supported.
gps	Indicates that coordinates in the file are GPS latitude and longitude (and optional altitude in meters) values. The default assumption, if "gps" is not specified, is that the coordinates are Cartesian coordinates in units of meters (or at least a consistent unit for the intended purposes).
[mmf | sdt]	These mutually-exclusive options set the desired output format that is directed to STDOUT.
speed <factor>	This option applies to the SDT format only. The given <factor> is used to alter the time scale used in the SDT "wait <msec>" commands that pace the rate of motion. For example, a "speed <factor>" of 10.0 causes the output SDT to contain node position updates that will playback at a rate of 10 times faster than real-time in the *sdt* or *sdt3d* visualization tools. The default <factor> is 1.0 which corresponds to normal, real-time execution.
range <commsRange>	When this option is given, the distance between nodes is compared to the given <commsRange> (in units of meters or Cartesian equivalent). Nodes that are separated by less than the <commsRange> are considered "linked" (or for MMF outputs a path loss of 0.0 dB is used). Nodes that are separated by a greater distance are considered "unlinked" (or for MMF outputs a path loss of 999.0 dB is used). If the "range" option is _not_ specified, then radio frequency (RF) path loss computation is used to determine node connectivity (see below). This computation and the parameters that it uses will be exposed in future versions of this tool. Meanwhile, the built-in defaults of 2.4GHz frequence, 225mW transmit power, -80 dBm receiver threshold with a 6 dB link margin are used. A Two-Ray path loss computation is used for "ground-to-ground" nodes that are sufficiently separated while a Free Space path loss computation is used for other types of connections. In the SDT format, the receiver threshold determines the "link" status while for the MMF format, the computed path loss values are given in the output.
power <milliwatts>	This specifies the transmit power (in milliwatts) used when the computation of graph connectivity is based on RF path loss calculation (i.e. "range" not specified). The default transmit power value is 225 mW.
frequency <hertz>	This specifies the transmit frequency (in Hertz) used when the computation of graph connectivity is based on RF path loss calculation (i.e. "range" not specified). The default transmit frequency value is 2.4 GHz.
height <meters>	This specifies the antenna heights (in meters) of the transmitter and receiver stations used when the computation of graph connectivity is based on RF path loss calculation (i.e. "range" not specified). The default antenna height is 0.0 meters.
dem <demFile>	This specifies the name of a file to use as a source of digital elevation model (dem) data (terrain profile). When this option is specified, the RF path loss computation uses the NTIA Irregular Terrain Model (ITM, aka Longley-Rice) algorithms that consider transmitter/receiver location, antenna heights and the terrain. Note that the <demFile> MUST included the full geographic area that the EmulationScript references. The <demFile> MUST be in a format that is supported by the Geospatial Data Abstraction Library (GDAL) code used for parsing terrain data files. For example, file formats such as DTED, GeoTIFF, and others are supported. Terrain data is readily available from the U.S. Geological Survey (USGS) "Seamless Server" that can be accessed via the Internet at http://seamless.usgs.gov. It is

	RECOMMENDED that the highest available resolution data (e.g. 1 arc-second or better) be used for the best results.

When the "`range`" option is specified, the resultant graph produced by the *gb* tool is connected based on a comparison of nodes' locations and the given `<commsRange>` value. When the "`range`" option is used, all of the RF parameters (`power`, `frequency`, etc) are ignored.

When the "`range`" option is NOT specified, then a computation of RF path loss is used to determine graph connectivity. In this case, the receive power (`rxPower`) is computed as "`txPower - pathLoss`" where the "`txPower`" can be set with the *gb* "`power`" command-line option and the "`pathLoss`" is computed based on the nodes' scripted locations. By default, the RF path loss computation is based on a hybrid approach of a Two-Ray and Free-Space calculation where the Two-Ray model is used for long distance (dependent on antenna height) ground<->ground links and the Free-Space equation is used for short range ground<->ground links, air<->ground links, and air<->air links. Note a node is considered an "air" node when its height is greater than 30 meters.

As an alternative to the default, hybrid Two-Ray/Free-Space model of RF propagation, the NTIA Irregular Terrain Model (ITM, aka Longley-Rice) of RF propagation can be invoked by specifying a file containing digital elevation model (terrain data) with the "`dem`" command-line option. The configured RF parameters (power, frequence, etc) are used with this model. Currently a number of defaults are assumed for other parameters this model accepts (e.g., polarity, "radio climate", dielectric constant, ground conductivity, etc). A future version of this program may provide command-line options to set these additiona RF model parameters. The current assumed defaults are given in the table below.

Table 4. gb ITM Model Parameter Assumptions

Parameter	Assumed Value
Polarity	HORIZONTAL
Radio Climate	CONTINENTAL TEMPERATE
EPS Dielectric	15
SGM Conductivity	0.005
Atmosphere Surface Refractivity	301
Time Variability	90%
Situation Variability	90%

After the RF path loss and resultant "rxPower" is computed, the "rxPower" is compared to a "receive power threshold" and "link margin" value. The current "receive power threshold" is -80dBm and the "link margin" is 6 dB. Thus, the "rxPower" must be greater than -74dBM for the nodes to be considered connected (i.e., "linked" in SDT terms). Note that when the Mitre Mobility Format (MMF) output format is specified, the computed path loss value is written to the output file along with the node identification and location information.

4.1.1. MITRE Mobility Format (MMF) Notes

The MITRE Mobility Format (MMF) is a simple (but potentially quite bulky) text file format that has been used to convey mobile node positions and pre-computed radio frequency (RF) propagation path loss and distances among the nodes. Each line of the text file is described as follows:

```
<time> <txNodeId> <rxNodeId> <loss>[/<~loss>] <dist> <txUTMx> <txUTMy> <txUTMz> <rxUTMx>
<rxUTMy> <rxUTMz>
```

The `<time>` is in integer units of seconds. The `<txNodeId>` and `<rxNodeId>` are integer node identifiers. The `<loss>` is a floating-point value that gives the path loss (in dB) from sender (*txNode*) to the receiver (*rxNode*). This path loss is assumed to be bi-directional unless the optional comma-delimited `<~loss>` is given which specifies a different path loss value from the *rxNode* to the *txNode*. The `<dist>` specifies the pre-computed distance between the *txNode* and *rxNode* locations. Finally, the remainder of the line specifies the *txNode* and *rxNode* locations, respectively, in Universal Transverse Mercator (UTM) coordinates with the "*UTMz*" value indicating altitude in meters.

Here is an excerpt from an example MMF file:

```
0 1 2 78.195327 90.132864 325900 4299310 0 325974 4299259 0
0 1 3 80.013243 100.076260 325900 4299310 0 325898 4299210 0
0 1 4 100.210723 1018.440683 325900 4299310 0 325056 4298972 460
0 1 5 95.185714 571.064644 325900 4299310 0 325593 4299167 460
0 2 3 78.197395 90.143596 325974 4299259 0 325898 4299210 0
0 2 4 100.600519 1065.186205 325974 4299259 0 325056 4298972 460
0 2 5 95.671124 603.887149 325974 4299259 0 325593 4299167 460
0 3 4 99.943938 987.635097 325898 4299210 0 325056 4298972 460
0 3 5 94.910163 553.232589 325898 4299210 0 325593 4299167 460
0 4 5 95.185686 571.062809 325056 4298972 460 325593 4299167 460
0 4 1 100.145221 1010.789275 325059 4298982 460 325898 4299309 0
0 4 5 95.223276 573.539590 325059 4298982 460 325603 4299164 460
1 2 1 78.066028 89.464490 325970 4299256 0 325898 4299309 0
1 2 3 78.197399 90.143615 325970 4299256 0 325894 4299208 0
1 2 4 100.485118 1051.127786 325970 4299256 0 325062 4298992 460
1 2 5 95.463742 589.639650 325970 4299256 0 325614 4299161 460
1 3 1 80.227050 101.315585 325894 4299208 0 325898 4299309 0
1 3 4 99.826103 974.327023 325894 4299208 0 325062 4298992 460
1 3 5 94.711128 540.699540 325894 4299208 0 325614 4299161 460
1 4 1 100.095321 1004.999068 325062 4298992 460 325898 4299309 0
1 4 5 95.267188 576.446505 325062 4298992 460 325614 4299161 460
1 5 1 95.026092 560.665946 325614 4299161 460 325898 4299309 0

...
```

This excerpt shows the node locations and interconnecting path loss, etc for 5 nodes at times 0 sec and 1 sec. Although this format is somewhat bulky for large scenarios, it has been useful for driving emulation scenarios with pre-computed (sometimes with very hi-fidelity RF propagation models) path loss information for the nodes in motion. The linearity with respect to time allows the file to be incrementally parsed as the emulation (or simulation) is executed.

5. PlanBuilder (*pb*) Tool

The PlanBuilder (*pb*) tool is an ancillary utility that allows existing SDT or *ns-2* files to be converted into a corresponding *MotionPlan* XML document. The resultant document can, in turn, can be used with the other tools described here to refactor a pre-existing scenario in one of these formats into an *EmulationScript* or other desired format. An interesting option in the *pb* tool is its "`origin`" option that enables conversion of scenarios from Cartesian->GPS or GPS->Cartesian coordinates.

5.1. Usage

The usage of the *pb* tool is summarized by:

```
pb input <inFile> [gps][origin <lat>,<lon>]
```

The required `<inFile>` parameter may be an SDT file or a text file containing *ns-2* node motion commands (e.g. "`set $node X_ <val>`", "`$ns at <time> '$node setdest xVal yVal [zVal] <velocity>'`", etc).

Table 5. pb command-line options

`input <inFile>`	Specifies the input filename `<inFile>` to be processed. SDT and ns-2 input files are currently supported.
`gps`	Indicates that coordinates in the file are GPS latitude and longitude (and optional altitude in meters). The default assumption, if "`gps`" is not specified, is the coordinates are Cartesian coordinates in units of meters.
`origin <lat>,<lon>`	When this option is given, the coordinates in the input file are remapped from Cartesian-to-GPS (default) or GPS-to-Cartesian coordinates in the *MotionPlan*

	that is output, depending if the "gps" input option is also specified. The specified <lat>,<lon> is assumed to correspond to a Cartesian (x,y) origin of (0.0,0.0).

The resultant *MotionPlan* is directed to the *pb* program's STDOUT. For example:

```
pb gps input file.sdt > motionPlan.xml
```

will generate an *MotionPlan* document named "motionPlan.xml" from the "file.sdt" input file.

The resultant *MotionPlan* will contain a set of MotionPlan:waypoints for each mobile node that is identified with locations, velocities, and durations that are computed by interpolating mobile node positions over time or direct use of *ns-2* "setdest" directives.

Note in the case of *ns-2* TCL scripts, leading '$' characters (that are used in dereferencing TCL variables) are removed and embedded numeric indices (e.g., as in the case of "$node(1)") are extracted and concatenated to construct a simpler mobile node "id" in the resultant *MotionPlan*. It is important to note that the related tools that output *ns-2* mobile node motion commands (e.g. the "*mg*" tool) similarly attempt to re-generate a valid TCL syntax by prepending node names with '$' and embedding any indices in the node name within parenthesis. For example, *pb* will convert an *ns-2* mobile node identified as "$node_(3)" to a MotionPlan:Node:id of "node_3". Similarly, the "*mg*" tool, when generating *ns-2* output, will convert a Node:id of "node_3" into "*$node_(3)*". This is expected to work well for conventional *ns-2* mobile node scripts, but there may be cases where this approach will produce unexpected results.

6. MMF to SDT (*mmf2sdt*) Conversion Utility

The Mitre Mobility Format (MMF) to SDT (*mmf2sdt*) converter is an ancillary utility that parses existing MMF files and outputs a corresponding SDT file including node position and link status (linked/unlinked). This allows an existing MMF file to be displayed with the SDT visualization tool or manipulated with other tools that can use the SDT format.

6.1. Usage

The usage of the *pb* tool is summarized by:

```
mmf2sdt zone <num> input <inputFile> [threshold <pathLoss>]
```

The required <inFile> parameter MUST be an MMF file containing pair-wise node location and connectivity (path loss) information.

Table 6. pb command-line options

input <inFile>	Specifies the input filename <inFile> to be processed.
zone <num>	Specifies the UTM zone of the coordinates in the MMF <inFile>. A positive zone value indicates the northern latitudes while a negative zone value indicates southern latitudes. For example, the central East Coast of the United States is zone 18.
threshold <pathLossValue>	When this option is given, the coordinates in the input file are remapped from Cartesian-to-GPS (default) or GPS-to-Cartesian coordinates in the *MotionPlan* that is output, depending if the "gps" input option is also specified. The specified <lat>,<lon> is assumed to correspond to a Cartesian (x,y) origin of (0.0,0.0).

Processing an input MMF file results in SDT commands being directed to STDOUT. The location coordinates of the SDT "node position" commands are GPS <longitude>,<latitude>[,<altitude>] values. SDT "link" and "unlink" commands are also output to indicate pair-wise node connectivity, given the provided "threshold" <pathLoss> value in dB. Note, at this time, the "*mmf2sdt* utility does NOT optimize the use of "node posiition" or "link" and "unlink" commands. I.e., "node position", "link" and "unlink" commands are issued somewhat gratituitously. A future version of the "*mmf2sdt* utility will keep the internal node location and graph state necessary

to minimize SDT link status updates. SDT "`wait`" commands are also used to convey the pacing of any mobility and time changes in the file.

7. Example Usage

(TBD) Describe some example usages of these tools with an emphasis in how they are used in concert to assist in mobile network emulation and simulation experimentation.