MGEN - Multi GENerator

A Toolset used to generate real-time traffic patterns and perform IP network performance tests and measurements using UDP and TCP IP traffic

Overview

- Installation
- Usage
- Commands
- Scripts Usage
- Traffic Patterns
- Log File

Installation:

- **Mgen Package install on Ubuntu**

  `sudo apt-get install -y mgen`

- **Download the MGEN source code distribution from the GitHub Repository.**
  - **UNIX:** (The MGEN program can be built with gcc on most Unix platforms)

  - In the main folder, Update the submodules with `git submodule update --init`

  - Navigate to the `makefiles` folder and Select the appropriate Makefile.{os} for your operating system and run:

    `make -f Makefile.{os}`

  - **WIN32:**

  - Navigate to `mgen/makefiles/win32`

  - Open the Visual C++ `mgen{VSVersion}.sln` file appropriate to your VS version. (e.g. mgen-2008-espress.sln)

  - Type F7 to build the `mgen.exe` program.

  (The executable will be found in the directory Release or Debug from "`mgen/win32`").

MGEN Usage

The mgen version 5.0 program must currently be launched from a command-line.

```
mgen [ipv4][ipv6][input <scriptFile>][save <saveFile>]
     [output <logFile>][log <logFile>]
     [binary][txlog][nolog][flush][hostAddr {on|off}]
     [event "<mgen event>"][port <recvPortList>]
     [instance <name>][command <cmdInput>]
     [sink <sinkFile>][block][source <sourceFile>]
     [interface <interfaceName>][ttl <multicastTimeToLive>]
     [unicast_ttl <unicastTimeToLive>]
     [tos <typeOfService>][label <value>]
     [txbuffer <txSocketBufferSize>]
     [rxbuffer <rxSocketBufferSize>]
     [start <hr:min:sec>[GMT]][offset <sec>]
     [precise {on|off}][ifinfo <ifName>]
     [txcheck][rxcheck][check][stop]
     [convert <binaryLog>][debug <debugLevel>]
     [localtime <localtime>] [queue <queue>]
     [broadcast {on|off}] [logdata {on|off}]
     [loggpsdata {on|off}] [gpsfile <fileName>]
     [df {on|off}][analytics] [window <secs>]]
     [report] [suspend <flowId(s)>] [resume <flowId(s)>] [reset <flowId(s)>]
     [retry <count>[/<delay>]
     [epochtimestamp]
```

MGEN Usage

- **A simple way to run MGEN for monitoring ports 5000,5004,5005, and 5006 for received UDP traffic and log to a specific file "log.drc"**

```
mgen port 5000,5004-5006 output log.drc
```

- **We can create scripts that will be given in input to MGEN. [By default the output is stdout if not specified otherwise]**

`mgen input script.mgn` **To achieve first command's result, script.mgn would have the following code:** `LISTEN UDP 5000,5004-5006`

- **The "EVENT" command can be used to achieve equivalent operation with the command-line syntax**

```
mgen event "listen udp 5000,5004-5006" output log.drc
```

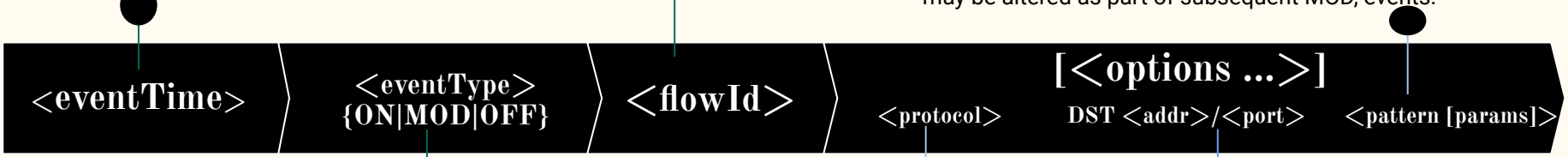
MGEN Script Format. *-Example with Transmission Events*

An instance of MGEN can simultaneously transmit traffic to multiple destinations with different patterns of transmission.

Each event is scheduled to be executed by MGEN at the relative time given by this field. If the field is omitted an <eventTime> of 0.0 (or immediately if MGEN is already started) is assumed.

Each Transmission event in the script is tagged with a <flowId>.

The generated traffic may vary in size and frequency, which can be defined in the <pattern> option. Each pattern is separately defined by its [params]. {available patterns: PERIODIC, POISSON, BURST, JITTER, CLONE}.
Note: Patterns must be specified in ON events and may be altered as part of subsequent MOD, events.



A transmission event represented by a flow has to be started with ON. Later in can be modified with MOD, and It has to be closed with OFF.

The <options> fields are used to describe the characteristics of each flows initiated. As an example we assume to define the protocol.
{currently supported: "UDP", "TCP", and "SINK"}

As a second option we define the Destination with DST, which must be followed by the <addr> (address) and the <port> (port).
Note: at the specified address/port an MGEN service must be active and listening.

Example:

This script line will originate a "flow" of MGEN UDP destined for the loopback address (IP address 192.168.1.100) port number 5000 beginning immediately when the script is executed. The messages will consist of 1024 byte messages at a regular rate of 1.0 per second:

```
0.0 ON 1 UDP DST 192.168.1.100/5000 PERIODIC [1.0 1024]
```

Generating MGEN Traffic through scripts

- **Receiving side**
  This script will be responsible for listening on the ports where the transmission side will be sending.

`Receiver.mgn` Is the script run on the receiving node. with the following code:

```
LISTEN UDP 5000,5004-5006
LISTEN TCP 6000,6004-6006
```

By running the script with the command:

```
Mgen input Receiver.mgn output receiver.drc
```

an Mgen process will be initiated that will be listening:

- On ports 5000, 5004, 5005, 5006 for an UDP traffic
- On ports 6000, 6004, 6005, 6006 for a TCP traffic

The information about the traffic will be saved be saved in a file receiver.drc (Content of a logfile will be described in the following slides)

Generating MGEN Traffic through scripts

- **Transmission side**
  This script will be responsible for describing how the traffic will be generated

`Generator.mgn` Is the script run on the generator node. with the following code:

```
0.0 ON 1 UDP DST 192.168.1.100/5000 PERIODIC [10.0 1024]
5.0 ON 2 TCP DST 192.168.1.100/6000 PERIODIC [20.0 1024]
10.0 MOD 1 DST 192.168.1.100/5004
15.0 MOD 2 DST 192.168.1.100/6004
20.0 MOD 1 POISSON [10.0 1024]
```

By running the script with the command:

```
Mgen input Generator.mgn
```

An Mgen process will be initiated and:

- Right after launching the script, an UDP traffic flow with ID 1, of 10 packets/second will start, where each packet is 1024 bytes. The DESTINATION node's address is 192.168.1.100 at the port 5000.
- Five seconds later, a TCP traffic flow with ID 2, of 20 packets/second will start, where each packet is 1024 bytes. The DESTINATION node's address is 192.168.1.100 at the port 6000.
- Ten seconds after the script is launched, the flow with ID 1 is MODIFIED, and the DESTINATION is changed to the port 5004
- Five seconds later, also the flow with ID 2 is been MODIFIED, and the DESTINATION port is changed to 6004
- Lastly, after twenty seconds the script is running, The packet transmission of flow 1 will be MODIFIED. It will now generate messages at an average rate of 10.0 messages per second following a Poisson (exponentially-distributed) interval.

Possible Patterns of Traffic Generation

PERIODIC: `PERIODIC [<rate>, <size>]`

Generates messages of a fixed <size> (in bytes) at a very regular <rate> (in messages/second)

POISSON: `POISSON [<Ave-rate>, <size>]`

Generates messages of a fixed <size> (in bytes) at at statistically varying intervals at an average <rate> (in messages/second)

BURST: `BURST [REGULAR|RANDOM <AveInterval> <PatternType> [<PatternParams>] FIXED|EXPONENTIAL <AveDuration>]`

This is used to generate Burst of other available Patterns (PERIODIC, POISSON or even BURST) with their parameters. The interval between on Burst and the next one can be REGULAR or RANDOM, and the duration of each Burst can be FIXED or EXPONENTIAL.

JITTER: `JITTER[<rate>, <size> <JitterFraction>]`

Generates messages of fixed <size> but with a varying <rate>, depending on the <JitteraFraction>. The JitterFraction defines the interval of deviation from the <rate>.

CLONE: `CLONE [<filetype> <filename>[<RepeatCount>]]`

We can also clone an available traffic pattern that was saved in a <filename> that has “tcpdump” as <filetype>, and we can specify how many time we want the pattern to be repeated with <RepeatCount>.

Log File in MGEN

The log file is used mostly at the receiving end to store the information about the received packets.
Each Line of the MGEN text log file format corresponds to a unique event and follows a particular convention as described below:

```
<eventTime> RECV proto> <protocol> flow> <flowId> seq> <sequenceNumber> src> <addr>/<port> dst><addr>/<port> sent> <txTime> size> <bytes> [host> <addr>/<port>] [gps> <status>,<lat>,<long>,<alt>] [data> <len>:<data>] [flags><flag>]
```

The `<eventTime>` corresponds to when the message was received. . The `<protocol>` specifies the protocol (udp,tcp,sink).The `<flowId>`, `<sequenceNumber>`, `dst><addr>/<port>`, and `<txTime>`, are from the payload of the MGEN message. "`dst`" corresponds to the destination address to which the source addressed the MGEN message. `<txTime>` and `<eventTime>` have the following format <hr:min:sec>GMT. `src> <addr>/<port>` is the source address. `size> <bytes>` correspond to the packet payload size. A maximum size of **65535** for each TCP fragment is set, if the message size exceed the limit the `flags>` field will be set to 0x01 to indicate that the message is incomplete, while the last fragment of the message will be signaled with the 0x02 flag. The `[host> <addr>/<port>]` corresponds to the MGEN message source's "perceived" default local address, It's an optional field and may not be present if the information is not valid. The global positioning system (GPS) information is available when the MGEN message source is used in conjunction with the NRL gpsLogger program. The optional "`data>`" attribute is present only if the received MGEN message contains optional user-defined payload.

References:

- https://github.com/USNavalResearchLaboratory/mgen
- https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/MGEN/
- https://github.com/USNavalResearchLaboratory/mgen/blob/master/doc/mgen.pdf