FlowMonitorMeasurer Group

Flow Monitor Measurer has several parameters, which define how it detects the target traffic flow.

Table of Contents

1. flowmap_interval

Determines how often the flow map is collected. A lower value consumes more resources but is more responsive.

- Unit: `milliseconds`
- Precision: `integer`
- Minimum: `50`
- Default: `1000`

> 📝 Setting this value below `200` is not advisable in production environments or regular use.

Example

To make Scopemon collect flow results twice per second (i.e., every 500 ms), define this parameter as:

```
[FlowMonitorMeasurer]
flowmap_interval=500
```

2. flow_timeout

The duration in which a flow can remain inactive before the measurement is stopped. Setting this value too low may result in duplicate measurement for a single traffic flow. Setting this value too high results in measurement carrying unnecessarily long after the flow has ended.

- Unit: `seconds`
- Precision: `integer`
- Minimum: `1`
- Default: `10`

Example

To set the flow timeout to 5 seconds, define this parameter as:

```
[FlowMonitorMeasurer]
flow_timeout=5
```

3. operation_mode

Flow monitoring can be set to operate either in single flow or multi-flow mode. In single flow mode, Scopemon expects to detect a single flow at a time and create a measurement for each flow. The detection of more than 1 flow results in a warning. In multi-flow mode, Scopemon carries out a measurement as long as one or more flows are detected.

- Values:
  - `0` Multi-flow mode - Detect and measure multiple flows
  - `1` Single flow mode - Detect and measure a single flow
- Default: `0`

Example

To switch to single flow mode, define this parameter as:

```
[FlowMonitorMeasurer]
operation_mode=1
```

4. probe_hostname

The hostname of the Probe which is used for flow monitoring. This can be omitted if the Probe is located on the same device where Scopemon is used.

- Type: `string`
- Default: `127.0.0.1`

Example
If the Probe is installed in another device at IP address *192.168.1.43*, define this parameter as:

```
[FlowMonitorMeasurer]
probe_hostname=192.168.1.43
```

5. probe_interface_index

Capture interface for flow monitoring. Typically `0` is the default OS interface.

- Precision: `integer`
- Minimum: `0`
- Default: `0`

Example
If the capture interface index is 2, define this parameter as:

```
[FlowMonitorMeasurer]
probe_interface_index=2
```

6. probe_port

The port number of the local Probe. This can be typically omitted unless the port where Probe serves control connections has been changed in Probe configuration.

- Precision: `integer`
- Minimum: `0`
- Maximum: `65535`
- Default: `8177`

Example
If Probe is configured to serve control connections on port 9776, define this parameter as:

```
[FlowMonitorMeasurer]
probe_port=9776
```

7. packet_filter

Packet filter is one of the most important parameters, as it defines which traffic flows are measured. The packet filter needs to be strict enough so that no irrelevant flows are captured. Otherwise the application is unable to select a flow for monitoring and throws out a flow unambiguity warning. For more information, see [Packet Filters in Qosium](#).

- Type: `string`
- Default: `ip`

The packet filter in this *FlowMonitorMeasurer* group can differ from the [measurement filter](#). Thus, set a filter in the *Measurement* group as well; the same than here or one that you want to use in the measurement phase.

> 📝 In versions before 1.6.0.0, the filter defined here also becomes the measurement filter when the packet_filter_mode in the Measurement group is set manual.

Example
To enable monitoring only for UDP traffic going through ports 6889 or 6890, define this parameter as:

```
[FlowMonitorMeasurer]
packet_filter=udp port 6889 or udp port 6890
```

8. reconnect_interval

If a connection cannot be established to the local Probe, Scopemon waits for a duration specified by this parameter and then attempts to reconnect.

- Unit: `milliseconds`
- Precision: `integer`
- Minimum: `0`
- Default: `1000`

Example
To attempt a reconnection after 500 milliseconds, define this parameter as:

```
[FlowMonitorMeasurer]
reconnect_interval=500
```

9. use_promiscuous_mode

Promiscuous mode allows the detection of incoming traffic that is not directed to the selected network interface. This scenario is common when capturing mirrored traffic, e.g., from a switch.

- Values:

- `true` - Allow detection of all incoming traffic
  - `false` - Allow detection of incoming traffic destined only for this interface
- Default: `true`

Example
To disable detection of traffic not designated to the network interface, define this parameter as:

```
[FlowMonitorMeasurer]
use_promiscuous_mode=false
```

10. user_id

User ID can be used to identify a controller, i.e., the Qosium Scopemon instance in this case. You can set this freely. The set value will appear in the results, where it can be used as a parameter to find results. Thus, you can use this as you wish as an identifier for your measurement, e.g., in a large-scale measurement setup. The User ID here applies only for the flow monitor measurement.

- Precision: `integer`
- Minimum: `0`
- Maximum: '4294967295'
- Default: `0`

Example
To set an id of 6 for this client, define this parameter as:

```
[FlowMonitorMeasurer]
user_id=6
```

11. write_date_code_format

Date code format governs the frequency of file creation when `write_multiple_files`. Whenever Scopemon detects a change in the date code, it automatically triggers new result files. A timestamp with this date code is then appended to the filename.

- Type: `string`
- Default: `yyyyMMdd`

Example
To write results every hour, define this parameter as:

```
[FlowMonitorMeasurer]
write_multiple_files=true
write_date_code_format=yyyyMMdd-hh
```

12. write_filename_suffix

File suffix string when forming a filename for measurement result files.

- Type: `string`

- Default: Empty

📝 This option has no impact when *write_flowmap_to_file* is set to *false*

Example

If defined for example as "MyMeasurement", the resulting filename is "flows_MyMeasurement.txt".

```
[FlowMonitorMeasurer]
write_flowmap_to_file=true
write_filename_suffix=MyMeasurement
```

13. write_flowmap_to_file

When true, flow measurement results are written to file.

- Values:
  - `true` - Results are written to file
  - `false` - Results are not written to file
- Default: `false`

Example

```
[FlowMonitorMeasurer]
write_flowmap_to_file=true
```

14. write_multiple_files

When true, flow results are written to multiple files. By default, one file is created for each day. For configuring multiple file writing frequency, see [write_date_code_format](#).

- Values:
  - `true` - Results are written to multiple files
  - `false` - All results are written into a single file
- Default: `false`

📝 This option has no impact when *write_flowmap_to_file* is set to *false*

Example

```
[FlowMonitorMeasurer]
write_flowmap_to_file=true
write_multiple_files=true
```

15. write_path

Set to override the path where measurement result files are stored. Use `/` as the directory separator.

- Type: `string`
- Default: Scopemon root directory

> 📝 This option has no impact when *write_flowmap_to_file* is set to *false*

Example

```
[FlowMonitorMeasurer]
write_flowmap_to_file=true
write_path=c:/temp
```