175 Cabot Street – Suite 210
Lowell, MA 01854 USA
www.cspi.com
NASDAQ: CSPI

Can Sniffer10G aggregate or merge ports?

Model:

N/A

Software:

Sniffer10G

Operating System:

N/A

Information:

Yes, using the Sniffer10G software, traffic from two or more ports, from the same or different Myri-10G adapters, can be aggregated into a single logical receiver.

SNF_FLAGS is the environment variable that controls process-sharing (0X1), port aggregation (0X2), and package duplication (0X3).

SNF_FLAGS=0X2 (Port aggregation or merging))

Flag 0X2 says that the port number that is passed to an application is actually a mask of port, not just one port.

```
export SNF_FLAGS=0x2
```

For example, when using tcpdump:

```
export SNF_FLAGS=0x2

env SNF_FLAGS=0x2 /path/to/tcpdump -i snf3
```

Sniffer10G aggregation is done in software by the library, not at the NIC level. Thus, when this option is enabled there is more software overhead due to preserve packet ordering (among a few other things).

The API function **snf_ring-recv_many** is not supported with aggregation (it will return EINVAL). The reasons for this are many: foremost, since the aggregation is done in software, the interface incurs more overhead with respect to the metadata that is necessary to release buffers once they are returned to the library. Even aggregation aside, there is no performance advantage to using recv_many instead of the regular receive—we only added it by user request.

The SNF_F_RX_DUPLICATE (0X3) packet duplication feature/flag can be used to send all traffic to all X application instances, where SNF_NUM_RINGS=x, whether they are snort, suricata, bro or whatever, but this is not a high performance implementation due to the overhead of copying data. Or if the duplicate flag is not used, all traffic is split with RSS. However, there is currently no way to use Sniffer10G to share all traffic among one group of rings and also

share all the traffic among another group of rings. You would need to mirror the traffic to a second myri10ge port and start the second application on the second port.

Users can ensure that a Sniffer10G-aware Libpcap library is linked to the application by setting SNF_DEBUG_MASK=0X3 in the environment when opening the Sniffer10G device snfX. For example:

```
$ SNF_DEBUG_MASK=0x3 SNF_FLAGS=0x2 /path/to/tcpdump -i snf3
```

Setting this SNF_DEBUG_MASK variable causes the SNF API to output memory mapping information when the Sniffer10G devide (snfX) is opened by Libpcap. If no information is dumped to the screen, it is likely that your application is linked against a version of libpcap that is not SNF aware. Monitor myri_counters to verify that traffic is in fact being received.

Revision	**Date**	**Change**
1	7/5/2016	Initial Draft
2	8/19/2016	Feedback Edits