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

While running in Sniffer mode, doing passive capture, does the device respond to non-multicast traffic (for example, IGMP)?

Model:

ARC Series C adapters (10G-PCIE2-8C2-2S)

Software:

Sniffer10G

Operating System:

Supports both Windows and Linux operating systems.

Information:

In Sniffer mode, (at least one process has done a **snf_open+snf_start**), all packets are sent up to the host (irrespective of unicast, multicast, broadcast, etc.).

Often, there is a requirement that some received packets are acknowledged. We have a solution for this exact problem- it is covered through the reflect to netdev functionality in the Sniffer API (with the function snf_netdev_reflect). If you identify an IGMP (or other packet) that you would have wanted the kernel to handle, you can “reinject” it up the kernel stack to be processed. Despite the fact that Sniffer is doing passive capture, the kernel still has a valid send and receive queue. The adapter doesn’t push anything in the receive queue while you are Sniffing but it can still put packets on the wire (like IGMP replies) from its send queue.

So, again, while Sniffer is handling high packet rate incoming multicast/UDP, the Sniffer also supports (with the netdev reflection capability) having the Ethernet driver handle all non-multicast traffic (presumably, IGMP).

For more details on the snf_netdev_reflect function, please see the **Sniffer10G API Documentation**, found in the Sniffer10G release. The release also includes several examples (**snf_multi_recv.c** and **snf_simple_recv.c**) of using this function in the share/examples directory.

Please note that for this functionality, your application must use the Sniffer10G API directly. This netdev reflection is not available when using the Sniffer10G Libpcap interface.

Also note that the use of **snf_netdev_reflect** is not meant for high performance. When packets are reflected they are copied back into the kernel via an ioctl and then send to the Ethernet interface. It is very inefficient and not meant for high performance. We were able to sustain 100Kpps 500-byte packets by using snf_simple_recv-N and then syncing the packets from another application. However, packet loss may occur even at lower rates.

Draft	Date	Change
1	7/11/2016	Initial Draft
2	7/22/2016	Feedback