Testing the Alloy NS-16J Switch Using Tcpdump

Centre for Advanced Internet Architectures.
Technical Report 031217A
Swinburne University of Technology
Melbourne, Australia
Ana M. Pavlicic, Grenville Armitage
apavlicic@swin.edu.au, garmitage@swin.edu.au

Abstract - **This technical report investigates how the Alloy NS-16J 16-port switch handles packets arriving on its backplane using three different methods of starting packet bursts. The packet bursts are generated by Netcom SmartBits2000 using Netcom SmartWindow software. The switch will be used in future research as part of the MAGIC project.**

Keywords - switch, CAM, MAC address, packet, tcpdump, port

I. Introduction

This report investigates the performance of the Alloy NS-16J 10/100Mbps switch using a Netcom Systems SmartBits2000 device to generate test traffic. The SmartBits2000 is configured with four SX-7410B 100Mbps Ethernet cards connected via CAT5 UTP cable to the switch. The Smartbits2000 was used to generate UDP packet streams using the Windows-based SmartWindow software provided. SmartWindow enables the user to send packets from one card to another specific card(s) or to simply flood all card ports with packets.

SmartWindow was used to estimate the actual CAM table size and flood tcpdump with packets from the switch to look at packet burst patterns. The process was then repeated with a hub and a high performance Cisco switch for comparison.

II. Test Set Up

A. *Physical connections between devices used in the investigation and SmartWindow.*

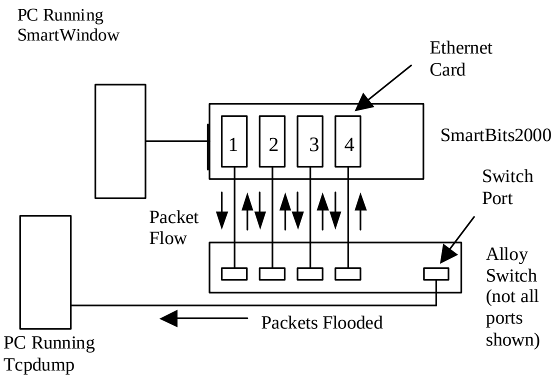

Figure 1 : Test Set up showing Tcpdump connection

For the CAM table tests each of the four cards were connected via CAT5 UTP to one port on the 16-port Alloy switch. In the tcpdump test, another UTP cable was connected to one of the switch/hub ports at one end and to a separate PC running tcpdump on the other. This set up can be seen in Figure 1.

With SmartWindow the user is able to set various test characteristics such as the link utilisation level, the total number of packets sent, the size of the packet payload, the number of different MAC addresses in the system, the protocol used etc.

III. CAM Table Size

A. *Investigating the CAM table size to determine whether the manufacturer's quoted size holds true.*

The manufacturer quoted the size of the NS-16J CAM to have room for "8k" MAC addresses [1]. The purpose of this test was to investigate the actual size of the CAM table to know its capability in handling a large number of differing MAC addresses.

B. *Can the CAM table hold 8,000 MAC address and port entries?*

The first step to finding the size of the CAM table was to fill the CAM with 8,000 MAC addresses using card 2. The utilisation was set to 1% on each card and packet payload size to 64 bytes. Card 1 was then used to cycle through the 8,000 MAC addresses as the destination of the packets it sent. 100,000 packets were sent. This test was repeated with the number of MAC addresses increased after every trial. It was found that flooding first occurred when 8,321 varying MAC addresses were sent by card 2 to the CAM table. This would suggest that the CAM was full at 8,320 MAC entries and did not record the 8,321st MAC address. This simple test determined the CAM to have enough room for 8,320 MAC addresses and their corresponding port entries. There were no packets lost during this test.

C. *Validating the size of the CAM table.*

This time card 3 was used to fill the MAC table with 8,000 MAC addresses and then cards 1 and 2 cycled though 4,000 MAC addresses, each sending 100,000 packets to card 3. The switch registered that the port card 3 was connected to as the source of all the MAC addresses, thus packets would then be sent to card 3. To avoid possible CRC errors, fragmented/undersized packets, card 1 cycled through the first 4,000 MAC addresses in the CAM table and card 2 through the last 4,000 MAC addresses. This test was repeated with the number of packets per card increasing until flooding occurred after 8,320 MAC addresses. To validate the results the test was repeated with card 4 filling the CAM table and cards 1, 2, and 3 sending packets to it. Again, there were no errors in these tests.

The next step of this investigation focused on repeating the tests above but this time instead of half (or a third) of the destination addresses being sent by each card, all two (or three) cards simultaneously cycled though the 8,000+ MAC addresses to the destination card. It was once again found that the CAM table could hold 8,320 MAC addresses before flooding occurred. Also, there were no CRC errors and/or fragmented/undersized packets recorded by SmartWindow.

IV. Flooding Tcpdump

A. *Using tcpdump to investigate packet collisions.*

This set of tests was to investigate whether starting the cards at different times would create any difference in packet error or loss. We hypothesized that errors could be caused by multiple packets arriving at the same time on the switch's backplane and colliding. By starting the cards at different times, the packets might not collide on the backplane.

For these tests the packet inter-arrival time was set to 200ms so as to ensure that the rate of packet arrivals is slow and gives enough time for packets to pass through the switch. The size of the packet payload was set to 1,024 bytes. All four cards flooded tcpdump with 10,000 packets each (see Figure 1, page 1) to send a total of 40,000 packets. This was so that tcpdump could record all packets that were sent out of the switch and measure the time intervals between these packets.

B. *Card Group Burst.*

In the first test, all four cards were started at the same time using the Group feature of SmartWindow. Table 1 below shows packet loss and CRC errors were recorded. Tcpdump only captured 38,714 packets from the switch.

	Card 1	Card 2	Card 3	Card 4
Packets Received	24,465	25,878	25,878	29,524
CRC Errors	0	0	4	13
Frag/Undersize	6	0	14	52

Table 1: Group burst

C. *Start All Cards Burst.*

In the second test instead of starting the Group feature of SmartWindow we used the option "Start All Cards". This option starts all cards with a 50 to 100ms lag between each card beginning to transmit. As we can see from Table 2, there were only very few errors, most likely because the packets did not converge on the switch backplane at the same time. Tcpdump captured 39,836 packets, the rest being dropped by the switch.

	Card 1	Card 2	Card 3	Card 4
Packets Received	29,918	29,917	29,835	29,835
CRC Errors	0	1	0	0
Frag/Undersize	6	6	0	0

Table 2: Start All Cards burst

D. *Manual Card Burst.*

The final test involved manually starting each card. This resulted in 1 to 2 seconds delay between each card starting to transmit packets. As can be seen in Table 3, there were no CRC errors, fragmented/undersized packets and all 40,000 packets arrived and were accounted for.

	Card 1	Card 2	Card 3	Card 4
Packets Received	30,000	30,000	30,000	30,000
CRC Errors	0	0	0	0
Frag/Undersize	0	0	0	0

Table 3: Manual burst

E. *Verifying timestamps of packets in the tcpdump file.*

Each tcpdump test above produced a tcpdump file that was consulted as to the reason for the packet loss behaviour. Since the alleged errors occur in the switch, not all packets arrived to the machine running tcpdump.

As can be seen in Figure 2, both the test run manually and with the "Start All Cards" option have a relatively even gradient, where the rate of packet arrivals over time is steady. It can be seen that the Group option graph does not have an even gradient, suggesting that the cumulative number of packets falls due to packets being dropped by the switch. Figure 3 shows a close up of the beginning of the test. Each point on the graph represents one packet. As seen for the Group and Start All graphs, four bursts of packets are followed by a 0.2 sec interval of time where no packets arrive. Also, the time interval between the Group packets within a burst is slightly smaller than the Start All option packets. The Manual graph clearly shows between 1 and 2 seconds several packets were sent by only one card, corresponding to the time it took to manually click "Start" on the second card.

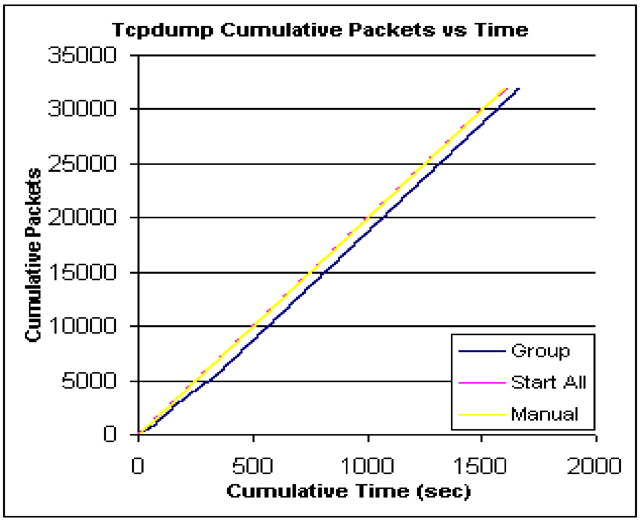

Figure 2: Cumulative packets vs time all start options (first 32,000 packets)

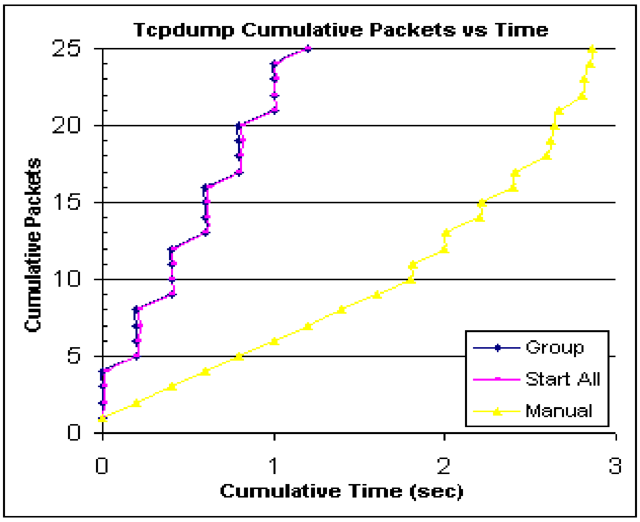

Figure 3: Initial packet bursts

Figure 4 shows a sample of packets burst using the Group option during a gradient change. We can see that before the gradient change occurs there are bursts of four packets and that the gradient change is caused by the tcpdump only receiving two packets per burst from the switch.

Figure 5 shows the cumulative packets versus cumulative time for the four individual source cards in the Group option test. We can see that the switch mainly dropped packets from card 4 with tcpdump only capturing 9,021 packets. Tcpdump captured 10,000 packets from card 2, 9,918 packets from card 1 and 9,775 packets from card 3. Figure 6 shows part of Figure 5, were some packet loss occured. These steps are caused by times when tcpdump was not recording four packets per burst as packet loss occured.

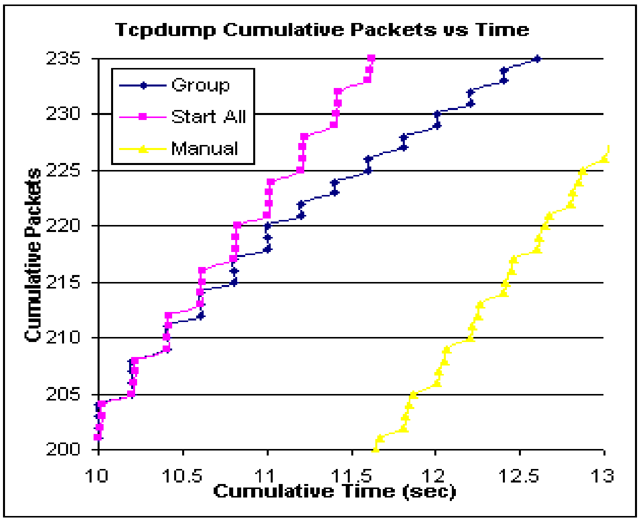

Figure 4: Gradient change in Group option

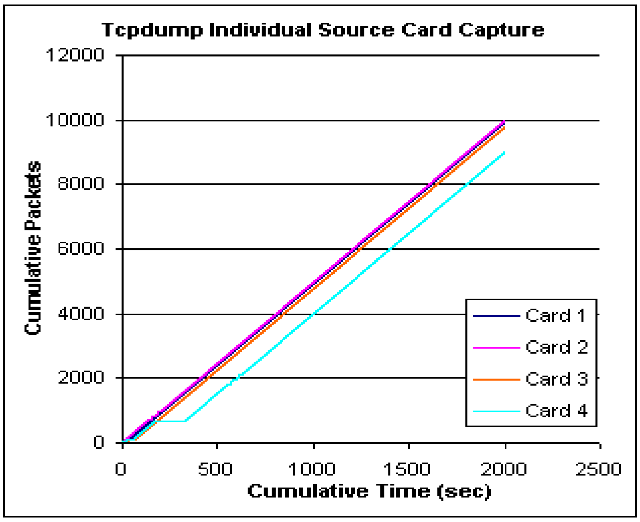

Figure 5: Group test individual card packet capture

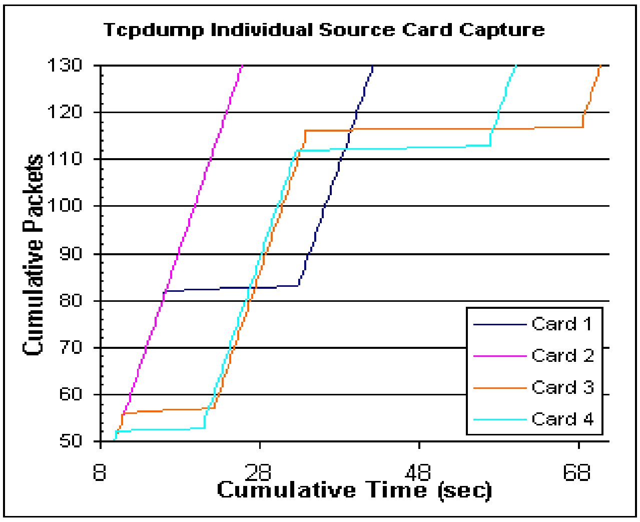

Figure 6: Gradient change due to packet loss in the Group test

F. Investigating the number of packets tcpdump would record when the Alloy switch is replaced with a hub.

This test used a 10Mbit/sec CentreCOM MR820TR hub in place of the Alloy switch with the same three tests (Group, Start All Cards and manual) run. In this case, not only did the hub result in CRC errors but also alignment errors and oversized packets rather than undersized/fragmented packets when the Group function was used. Once again, fewer errors occurred with the "Start All Cards" option and no errors or packet loss when the cards were started manually. The test showed that the same trend occurred in the hub as in the switch due to colliding packets.

F. *Confirming tcpdump was not responsible for packet loss by repeating the test using a high performance switch.*

To dismiss the possibility that the packet loss was a result of tcpdump failing to capture packets sent by the NS-16J, a Cisco Catalyst 2900 Series XL switch was tested while using the Group function in place of the Alloy switch. Cisco Discovery Protocol was disabled and all spanning-tree packets were filtered on all ports so as to ensure that the switch did not send out any broadcast packets not received by a SmartBits2000 card.

Graph 7 (note only the first 32,000 packets are shown) shows the culumative packets versus cumulative time when 40,000 packets were flooded by the four SmartBits cards (10,000 packets per card). All 40,000 packets were accounted for by tcpdump with no errors, proving that both the Catalyst 2900 and tcpdump were capable of handling all packets.

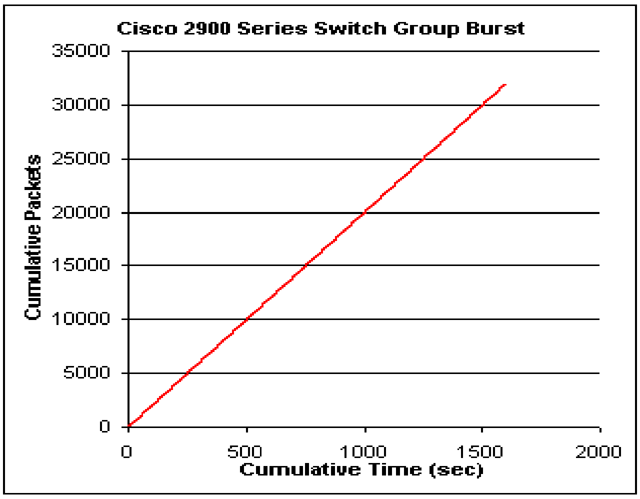

Figure 7: Group test using a Cisco Catalyst 2900 Series switch

Graph 8 shows a close up of the beginning of this test. Each dot represents one packet. As we can see the Catalyst 2900 was capable of handling bursts of four packets every 0.2 seconds from the beginning of the test to the end.

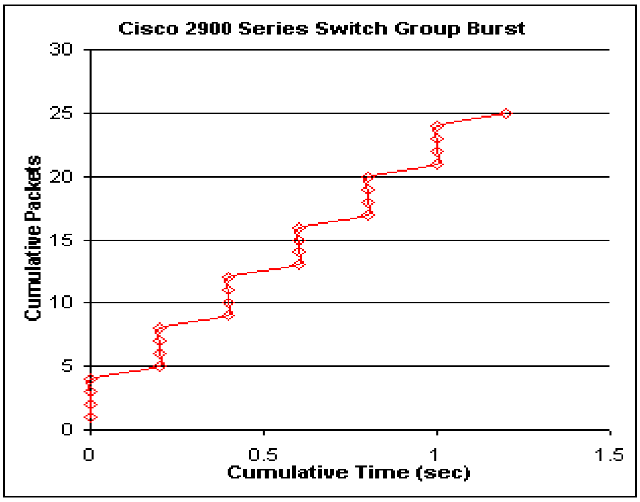

Figure 8: First 25 packets with the Cisco Catalyst 2900 Series switch

V. Conclusion

This investigation looked into the actual size of an Alloy NS-16J CAM table. It also tested the switch using three different packet burst start methods from four sources. The investigation found that the CAM table could hold 8,320 destination MAC address and port combinations, well above the manufacturer's quoted number of 8,000. It also found that bursting four packets onto the switch backplane at exactly the same time (or at very tiny inter-packet intervals) caused some errors and packets loss.

It was clear from the tcpdump test performed that the Alloy switch is not capable of handling loads such as those on high-speed networks and Internet backbones. Switches such as the Cisco Catalyst 2900 Series XL which would be more suited to this type of traffic demand where packets may simultaneously converge on the switch backplane. The Alloy NS-16J could, however, be adequate for small-scale projects where the rate and number of packets is significantly lower. This includes the MAGIC project at the Center for Advanced Internet Architectures.

It is important to note that the Cisco 2900 Series XL is substantially more expensive than the Alloy NS-16J. At the time of writing the Alloy NS-16J retails around the mid $AU100 range. The Cisco 2900 Series XL is no longer on the market. The next available Cisco switch is the WS-C2950-24 model around the mid $AU800 range. The Alloy switch is therefore more economical for smaller-scale projects. This difference in price as well as performace capabilities should be considered when purchasing any switch.

ACKNOWLEDGMENTS

Dr. Paul van den Bergen from the Centre for Advanced Internet Architectures, Swinburne University of Technology passed along his knowledge on operating the SmartWindow software package.

REFERENCES

[1] Alloy User Manual NS-16J and NS-24J, 16/24-port 10/100Mbps, Auto negotiation Switches (as of May 7, 2003).