

SNS Control Systems



A new Tool to study Network Stack Exhaustion in VxWorks

Epics Collaboration Meeting

Dec. 8, 2004

Sheng Peng

Ernest L. Williams Jr.

David Thompson

ICS _ Software Engineering Group

1



The Story



- When we were dealing with “IOC Disease” earlier this year we got pretty good at using vxWorks diagnostics tools, mBufShow, inetStatShow, and a few that WRS gave us like ifQValuesShow.
- We got pretty good at “tuning” by setting mbufs, driver queues, and the “if_Q length”.
- We found and fixed several causes of depleted buffers.
- **We still have errors!** Diagnostics like ifShow indicates txErrors and we still get white screens.
- The end driver with debugging turned on also reports txErrors.

The first round of cures



inetstatShow

Active Internet connections (including servers)

PCB Proto Recv-Q Send-Q Local Address Foreign Address (state)

...
1b4a990 TCP 0 **8184** 172.31.124.20.5064 172.31.124.107.51553 << Archive server

....

mbufShow

CLUSTER POOL TABLE

size clusters free usage

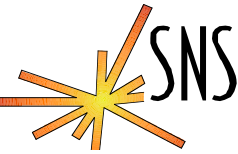
64 800 **772** 9859
128 1600 **1531** 105147601
256 800 800 2138545
512 400 400 34635
1024 200 200 1913
2048 300 300 27947
4096 20 20 6197

Eventually the archive server would consume all of the buffers because daily restarts never closed the old sockets. It usually took several days to a week for the IOC to crash, especially with large buffer configurations.

Other clients, were problems as well. Edm with a stuck mouse would do the same thing!

The net result is that we understand this and have fixed the problems with clients for the most part.

The second round



- **Now what? We still have problems and the IOCs have plenty of free buffers in the network stack.**
- **Maybe it is time to look at traffic patterns.**
- **Bring in etherreal!**

Network Traffic Analysis (Setup)



- **IOC Under Study:**

- » **Sci-hprf-ioc05 (without Beckhoff driver)**

- **Connected to CISCO 2950 layer2 switch**

- **lin-ics-netw3b1 ---- port 1**

- **Port 1 is mirrored for observation via a linux-based packet capture and analysis system.**

- **Tools used:**

- » **Laptop with “Ethereal” packet capture software**

- **NIC 1 (eth0) ---- used for remote access to the packet capture station**

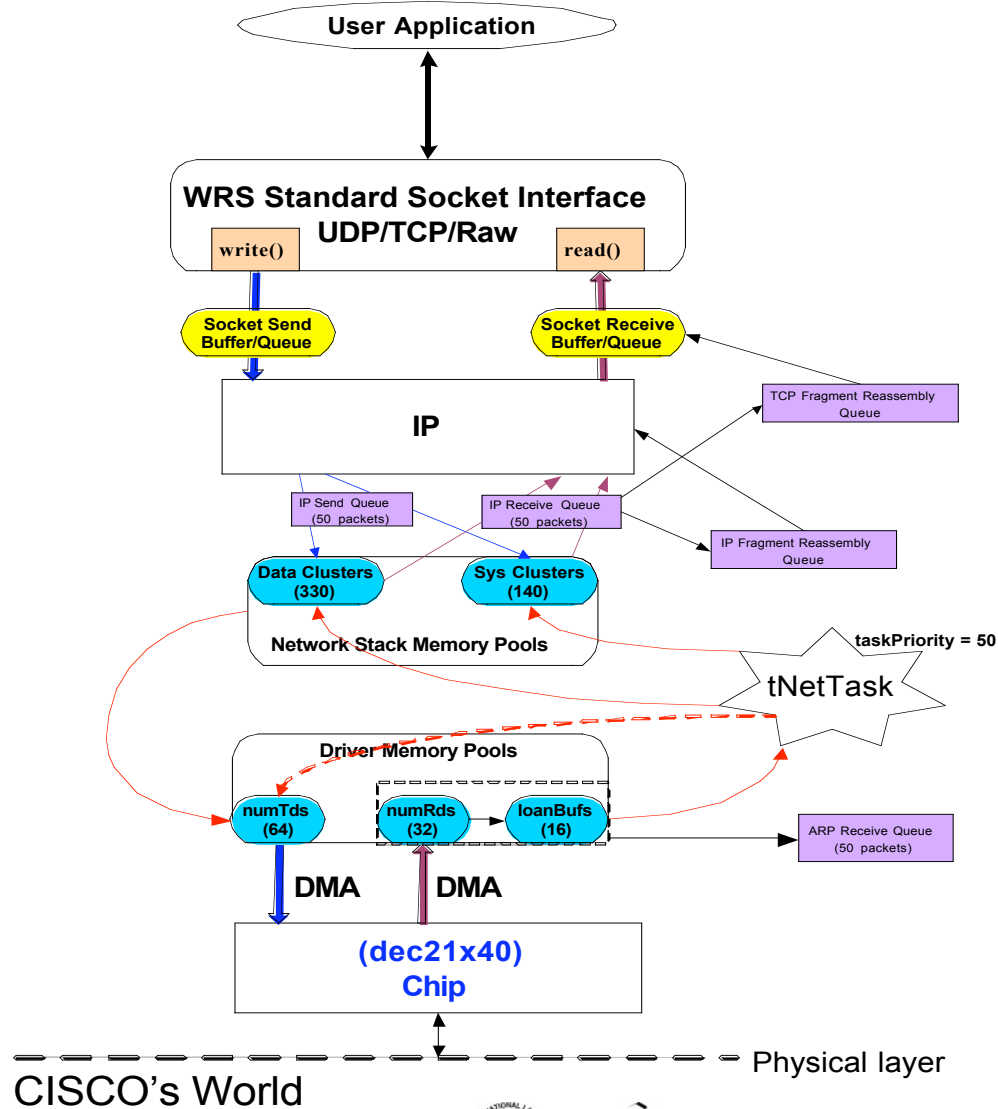
- **NIC 2 (eth1) --- connected to the CISCO port mirror.**

What we will study?



- **What we will study:**
 - » **Network Memory Resources Model**
 - What are mBufs?
 - What are mBlks?
 - What are cBlks?
 - What are clusters?
 - » **Flow diagram of the vxWorks Network Stack**
 - How are packets moved in out with respect to the OSI model?
 - » **The journey of a network packets as seen through the eyes of a network sniffer in an EPICS environment.**
 - We will make a timeline of events from the time an IOC is booted to when it is “open for business”
 - CISCO port auto-negotiation and turn-on
 - Loading of vxWorks image from boot server
 - Re-setting of IOC's network hardware by vxWorks
 - Loading startup file common to all vxWorks IOCs (i.e. common.cmd)
 - Loading application specific startup file (i.e. st.cmd)
 - loclnit
 - » **What protocols are showing up?**
 - Needed in the context of EPICS
 - Nuisance Protocols/traffic

VxWorks Network Stack Flow



Real-Time OS Considerations



- **Buffer management**
 - » Pre-allocated buffers as opposed to dynamic from the global heap at run-time
- **Timers**
 - » Connection management
 - » Timeouts
 - » Retries
- **Latency**
 - » Fast and deterministic interrupt handling interfaces
 - » Small thread context switch times
- **Concurrency**
 - » Smart use of semaphores
- **Minimized Data Copying**
 - » The TCP/IP implementation should minimize the amount of data copying. The data within each **frame** can be maintained in the **same buffer** so it doesn't need to be copied and re-copied by the CPU at each stage of the protocol. The networking chip's **DMA** places the packets directly in the **managed buffer pool** where the packet is passed up through the stack by manipulating **pointers** and not by copying data. The mbuf mechanism has been extended to allow the data to be shared between mbufs and mblocks where there are STREAMS protocols also present in the system.

Protocols we deal with in EPICS



- **UDP port 5065**
 - » CA beacons (“I am here” Heartbeat)
 - » Used to re-establish CA TCP virtual circuits
 - » CA beacons do not expect any replies
 - » The CA Beacon Daemon is listening on UDP port 5065.
 - A.K.A caRepeater
- **UDP port 5064**
 - » CA search message
 - » A response is expected within some timeout interval
- **TCP port 5064**
 - » CA server establishes a virtual circuit on port 5064
- **NFS**
 - » UDP port 111
 - Loading up IOC application
 - Running autosave/restore
 - Re-directing IOC files to boot server
- **NTP**
 - » UDP port 123
 - Keep IOCs time in synch
 - At SNS we should see this about every 10 seconds in our current configuration.
- **RSH**
 - » UDP port 514
 - Remote login support
 - Cat in vxWorks image

Network Traffic Analysis



Ethereal Packet Analysis Timeline

PowerOn IOC
T - 0 sec

Bring NIC online
T + 3 sec

EPICS neighbors come
T + 3.02 sec

| No. | Time | Source | Destination | Protocol | Info |
|-----|----------|----------------|--------------------|----------|--|
| 1 | 0.000000 | Cisco_dd:a8:41 | CDP/VTP | LLC | II, func=III; SNAP, OUI 0x000000 (Cisco), PID 0x2004 |
| 2 | 0.000740 | Cisco_dd:a8:41 | CDP/VTP | LLC | II, func=III; SNAP, OUI 0x000000 (Cisco), PID 0x2004 |
| 3 | 0.999483 | Cisco_dd:a8:41 | CDP/VTP | LLC | II, func=III; SNAP, OUI 0x000000 (Cisco), PID 0x2004 |
| 4 | 1.000225 | Cisco_dd:a8:41 | CDP/VTP | LLC | II, func=III; SNAP, OUI 0x000000 (Cisco), PID 0x2004 |
| 5 | 1.999654 | Cisco_dd:a8:41 | CDP/VTP | LLC | II, func=III; SNAP, OUI 0x000000 (Cisco), PID 0x2004 |
| 6 | 2.000412 | Cisco_dd:a8:41 | CDP/VTP | LLC | II, func=III; SNAP, OUI 0x000000 (Cisco), PID 0x2004 |
| 7 | 3.014213 | Cisco_dd:a8:41 | Spanning-tree (for | EAP | Success |
| 8 | 3.021231 | 172.31.80.16 | 255.255.255.255 | UDP | Source port: 1027 Destination port: 5055 |
| 9 | 3.022837 | 172.31.72.60 | 172.31.75.255 | UDP | Source port: 1029 Destination port: 5054 |
| 10 | 3.023061 | 172.31.72.60 | 172.31.75.255 | UDP | Source port: 1029 Destination port: 5054 |
| 11 | 3.024569 | 172.31.74.127 | 172.31.75.255 | UDP | Source port: 1029 Destination port: 5055 |
| 12 | 3.047139 | 172.31.72.2 | 224.0.0.2 | HSRP | Hello (state Standby) |
| 13 | 3.054543 | 172.31.72.64 | 172.31.75.255 | UDP | Source port: 1031 Destination port: 5055 |
| 14 | 3.090461 | 172.31.72.52 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 15 | 3.092848 | 172.31.72.41 | 172.31.75.255 | UDP | Source port: 5054 Destination port: 5055 |
| 16 | 3.106273 | 172.31.73.150 | 172.31.75.255 | UDP | Source port: 5054 Destination port: 5055 |
| 17 | 3.153825 | 172.31.72.70 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 18 | 3.155627 | 172.31.74.121 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 19 | 3.185355 | 172.31.72.37 | 172.31.75.255 | UDP | Source port: 5054 Destination port: 5055 |
| 20 | 3.189050 | 172.31.72.82 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |

Frame 1 (50 bytes on wire, 50 bytes captured)
IEEE 802.3 Ethernet
Destination: 01:00:0c:00:00:00 (CDP/VTP)
Source: 00:0c:85:dd:a8:41 (Cisco_dd:a8:41)
Length: 40
Trailer: 000000000000
Logical-Link Control
Data (32 bytes)

```
0000 01 00 0c 00 00 00 00 85 dd a8 41 00 28 aa aa ..... A.C.  
0010 03 00 00 0c 20 04 01 00 01 00 0b 73 5e 73 2d 53 .... . . . snr-c  
0020 73 00 00 02 00 05 03 00 03 00 05 a5 00 04 00 0a .....  
0030 00 0c 85 dd a8 41 00 00 00 00 00 00 ..... A. ....
```



Network Traffic Analysis (Cont'd)



Ethereal Packet Analysis Timeline

The screenshot shows a Wireshark interface with a list of NFS packets. A red arrow points from a yellow warning box to a specific packet in the list. The detailed view below shows the packet structure: Ethernet II, Internet Protocol, User Datagram Protocol, and Remote Procedure Call. The RPC details show a Network File System (NFS) LOOKUP Call. The warning box indicates that a retransmit is necessary.

| No. | Time | Source | Destination | Protocol | Info |
|-------|-------------|--------------|---------------|----------|--|
| 52943 | 31.9, 35424 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3308 v2 LOOKUP Call (Reply In 3309), DH:0a91304bc/RS-1-7 |
| 52950 | 31.9, 35432 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3310 v2 LOOKUP Call (Reply In 3311), DH:0a8d8333ac/bin |
| 52952 | 31.9, 35640 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3312 v2 LOOKUP Call (Reply In 3313), DH:0a725579ac/victorforke-ppc000 |
| 52954 | 31.9, 35696 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3314 v2 LOOKUP Call (Reply In 3315), DH:0a6a6517ac/rcf498_search |
| 52955 | 31.9, 35675 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3315 v2 READ Call (Reply In 3316), FH:0a5c1e23ac Offset:0 Count:8192 To |
| 52953 | 31.9, 35898 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3325 v2 READ Call (Reply In 3326), FH:0a5c1e23ac Offset:1100320 Count:8 |
| 52970 | 31.9, 36187 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3332 v2 READ Call (Reply In 3333), FH:0a5c1e23ac Offset:1100384 Count:8 |
| 52977 | 31.9, 36327 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3333 v2 READ Call (Reply In 3334), FH:0a5c1e23ac Offset:1589794 Count:8 |
| 52989 | 31.9, 36770 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3347 v2 READ Call (Reply In 3348), FH:0a5c1e23ac Offset:1597966 Count:8 |
| 52991 | 31.9, 37211 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3356 v2 READ Call (Reply In 3357), FH:0a5c1e23ac Offset:1598148 Count:8 |
| 52999 | 31.9, 37558 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3361 v2 READ Call (Reply In 3362), FH:0a5c1e23ac Offset:1598340 Count:8 |
| 52995 | 31.9, 38100 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3369 v2 READ Call (Reply In 3370), FH:0a5c1e23ac Offset:1522502 Count:8 |
| 52913 | 31.9, 38545 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3375 v2 READ Call (Reply In 3377), FH:0a5c1e23ac Offset:1530728 Count:8 |
| 52900 | 31.9, 38994 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3383 v2 READ Call (Reply In 3384), FH:0a5c1e23ac Offset:1530916 Count:8 |
| 52929 | 31.9, 39428 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3390 v2 READ Call (Reply In 3391), FH:0a5c1e23ac Offset:1547108 Count:8 |
| 52935 | 31.9, 39894 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3397 v2 READ Call (Reply In 3398), FH:0a5c1e23ac Offset:1550300 Count:8 |
| 52942 | 31.9, 40334 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3405 v2 READ Call (Reply In 3406), FH:0a5c1e23ac Offset:1550492 Count:8 |
| 52949 | 31.9, 40774 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3412 v2 READ Call (Reply In 3413), FH:0a5c1e23ac Offset:1571084 Count:8 |
| 52957 | 31.9, 41214 | 172.31.72.79 | 172.31.72.101 | NFS | RPC retransmission of #3419 v2 READ Call (Reply In 3420), FH:0a5c1e23ac Offset:1579276 Count:8 |

From 31.9 (152 bytes on wire, 152 bytes captured)
 Ethernet II, Src: 00:01:af:0a:25:19, Dst: 00:05:5b:3a:7b:0a
 Internet Protocol, Src Addr: 172.31.72.79 (172.31.72.79), Dst Addr: 172.31.72.101 (172.31.72.101)
 User Datagram Protocol, Src Port: nfs (396), Dst Port: nfs (2049)
 Remote Procedure Call, Type:Call KID:0a019e357
 Network File System, LOOKUP Call DH:0a91304bc/boot.lnx

Warning!!
NFS is heavy

Why is a retransmit necessary, hmmm?

```

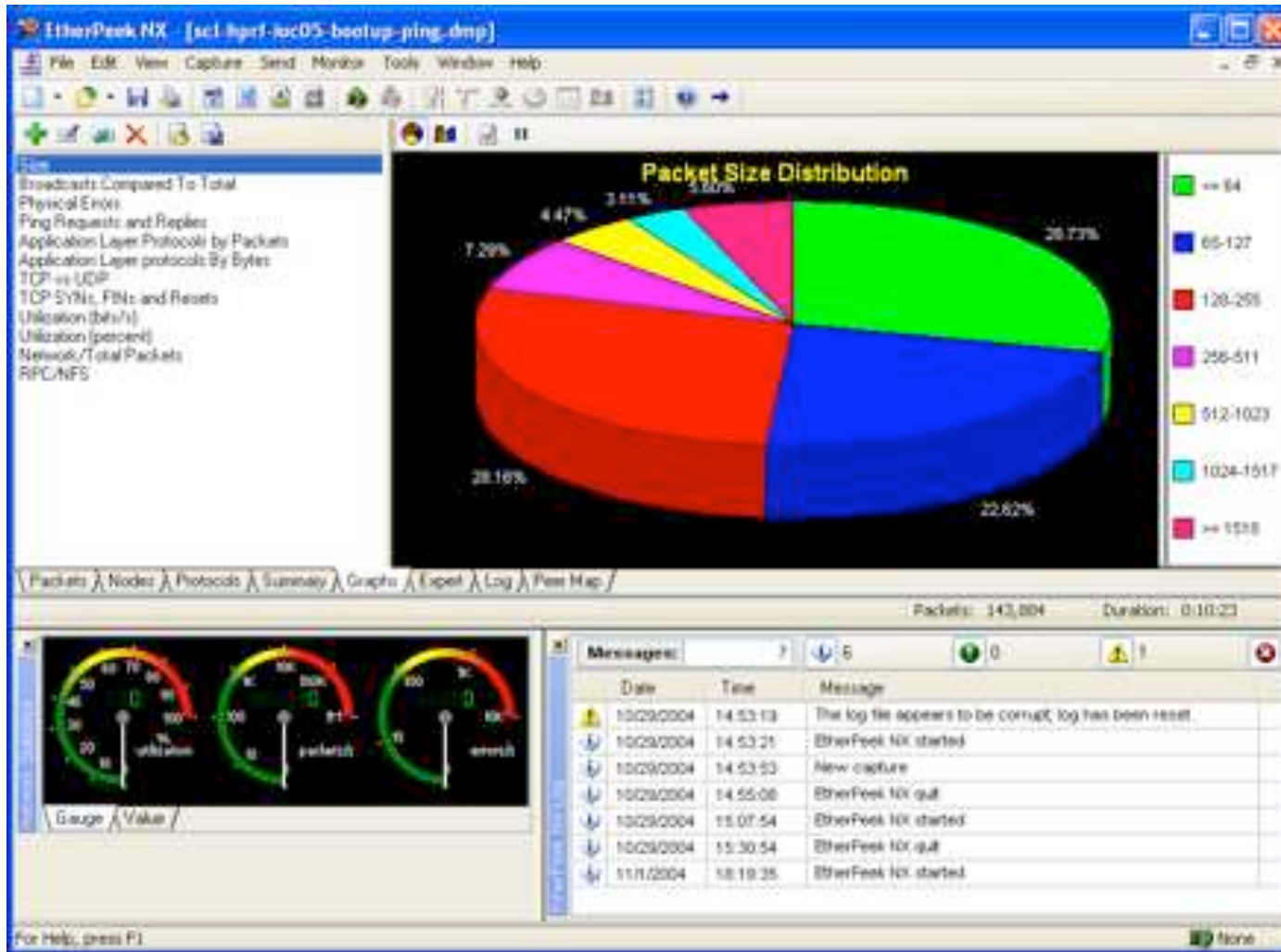
0000  00 05 5b 3a 7b 0a 00 01  af 0- 25 13 08 00 45 00  ..D.L...K..K.
0010  00 06 00 02 00 00 40 11  90 9c ac 1f 48 9c ac 1f  ..K.....K..K.
0020  49 5c 03 e2 09 01 00 80  2c 44 01 9e 83 57 00 00  ..K.....K..K.
0030  00 00 00 00 00 02 00 01  83 a3 00 00 00 02 00 00  ..K.....K..K.
0040  00 04 00 00 00 01 00 00  00 28 00 00 27 18 00 00  ..K.....K..K.
0050  00 00 00 53 53 73 24 53  53 5c 23 73 72 73 31 00  ..K.....K..K.
    
```

Network Traffic Analysis (Annotated)



Network Analysis (Packet Size Distribution) SNS

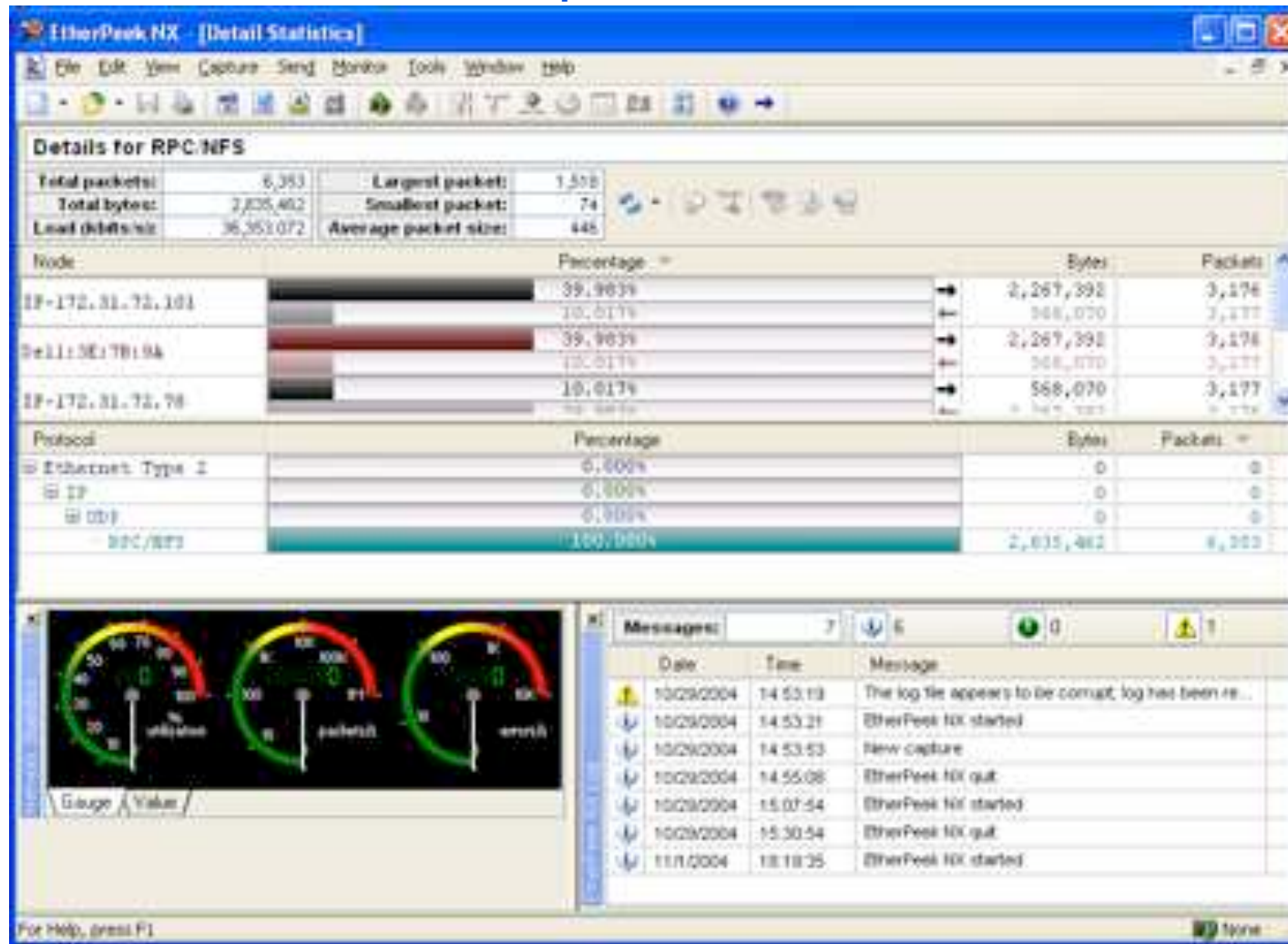
Sci-hprf-loc05



Network Analysis (NFS/RPC statistics)



Sci-hprf-ioc05



Network Analysis: Data Collection on Network Queues



PROTOCOL RECEIVE QUEUES

- **Healthy:**

- » dtl-llrf-ioc1a> protocolQValuesShow

- IP receive queue max size = 50

- IP receive queue drops = 0

- ARP receive queue max size = 50

- ARP receive queue drops = 0

- value = 28 = 0x1c

- **Unhealthy:**

- » scl-hprf-ioc05> protocolQValuesShow

- IP receive queue max size = 50

- IP receive queue drops = 107

- ARP receive queue max size = 50

- ARP receive queue drops = 0

- value = 28 = 0x1c

Network Analysis: Data Collection on Network Queues



IP SEND QUEUES

- **Healthy:**

- » dtl-llrf-ioc1a> ifQValuesShow("dc0")
dc0 drops = 0 queue length = 0 max_len = 100
value = 46 = 0x2e = '.'

- **Unhealthy:**

- » scl-hprf-ioc05> ifQValuesShow("dc0")
dc0 drops = 200 queue length = 0 max_len = 100
value = 48 = 0x30 = '0'

What can go wrong with the Network Stack?



- **Disruption of tNetTask via deadlock causing sockets not to be read.**
- **User tasks in general should have a priority lower than tNetTask. (i.e. greater than 50)**
- **Do not create and then take SEM_INVERSION_SAFE semaphores before making a socket call or your task could be promoted to run at tNetTask level**
 - » **tNetTask netTask 1cee480 0+I PEND**

What can go wrong with the Network Stack?



- **Application may have deadlock conditions which prevent them from reading sockets.**
- **If inetstatShow (or equivalent in other systems) displays data backed up on the send side and on the receive side of the peer, most likely there is a deadlock situation within the client/server application code.**
- **Running both server and client in the target by sending to 127.0.0.1 or to the target's own IP address is a good way to detect this kind of problem.**
- **Heavy NFS traffic may require an increase in driver memory pool.**

Results/Conclusions



- **The Network Analysis allows tuning of the network stack from apriori information as well as empirical data collected from the real environment.**
- **We have discovered some devices on our network that have improper configurations and hence cause unnecessary traffic.**
- **We have discovered that NFS is really a heavy hitter and that autosave/restore request files should be stored in one location.**
- **We have discovered that IGMP snooping must be supported on the CISCO edge switches to contain Allen Bradley Control Logix PLC multicast traffic. Multicast traffic should be contained in general.**
 - » **We moved from the CISCO 3500 series to the CISCO 2950 series**
 - CISCO 3500 series only supported CGMP snooping
- **We learned that sometimes IOC application errors are the main cause of Network Stack Exhaustion and/or failure.**
- **We have added an “open-source” network sniffer (Ethereal) to our EPICS Network trouble-shooting ToolKit.**
- **We have built in the Network diagnostics show routines from WRS in to our IOC’s common support library.**

Outline



- Introduction
- Implementing a network stack in the context of a Real-Time OS (RTOS)
- Basic Definitions and Memory Pools
- Network Stack Flow Diagram
- Network Traffic Analysis (w/ethereal)
- What can go wrong with the Network Stack?
- Results/Conclusion

Basic Definitions



Fundamental Data Structures

- **Mbufs (deprecated):**
 - » stores small stack data structures such as socket addresses, and packet data. Mbufs were designed to facilitate passing data between network drivers and the network stack, and contain pointers that can be adjusted as protocol headers are added or stripped. Mbufs contain space within them to store small amounts of data. Larger amounts of data were stored in fixed-sized clusters (typically 2048 bytes), which could be referenced and shared by more than one mbuf.
- **Clusters:**
 - » Network Data containers of various sizes in bytes
 - » Data containers must be a power of two
- **cBlks:**
 - » The cBlk is a structure that contains a pointer to the cluster data, the cluster size, and an external reference count. the "cluster block" was added, supporting the zbuf sockets interface and multiple network pools in addition to cluster sharing.
 - » One cluster block is required for each cluster
- **mBlks:**
 - » The mBlk is a structure that contains a pointer to a cBlk or another mBlk. mBlks are basically a modified version of the BSD style mbufs. The difference is that they now reference external clusters rather than carrying data directly. They are now called "mblocks."

The 3 main Network Memory Pools



- **Network Stack “Data” Pool:**

- » Data pools are used for packet send data with extra space for protocol headers. Clusters from the pool are allocated in the socket layer. The function which offers info about it is **netStackDataPoolShow()**. You can configure it with the definition of NUM_64, ... , NUM_2048.
- » application layer → network stack layer → network driver

- **Network Stack “System” Pool:**

- » System pools are used for network structures (sockets, routes, etc). The function to offer info about it is **netStackSysPoolShow()**. You can configure it with the definition of NUM_SYS_64, ... , NUM_SYS_512

- **Network “Driver” Interface Pool:**

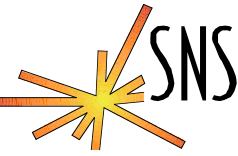
- » Buffer pool for each network interface. Data from the wire is received in clusters from a network device pool. These buffers are then passed up to the network stack. This pool is also used for staging packets to be transmitted by the target. The driver pool can be shown with the following utility routine: **endPoolShow(“dc”,0)** for our MVME2101 boards. Call **muxShow()** to show network driver info.

More on the Driver's Pool



- **Cluster size for ethernet is 1520**
 - » Cluster size has to be big enough to receive or transmit the maximum packet size allowed by the link layer. In this case that is 1518 bytes
 - » Two extra bytes are required to align the IP header on a 4 byte boundary for incoming data.
 - » Default number of clusters is 80.
 - » END network drivers lend all their clusters.
 - » **Clusters = numRds + numTds + NUM_LOAN**
 - Where numRds (32) is the number of receive descriptors
 - Where numTds (64) is the number of transmit descriptors
 - NUM_LOAN (16) is the number of loan buffers
 - mBlks = 4(numRds + NUM_LOAN)
 - Currently in the field for SNS06a and SNS06c we have:
 - numRds = 32, numTds = 64, and NUM_LOAN = 16
 - mBlks = 192, Clusters = 112
 - Should this be increased for some Apps? If yes, we need a configuration parameter in the “other” field.
 - Driver Pool for END drivers are configured in:
 - \$(WIND_BASE)/target/config/<bsp>/configNet.h

Basic Definitions (Cont'd)



Network Stack Queues

- **Queues are used to hold data waiting to be processed**
 - » Queues are implemented as a linked-list.
 - » Clusters are chained the to the queue's linked list
- **Types of Queues:**
 - » **Receive Queues:**
 - IP PROTOCOL RECEIVE QUEUE
 - FRAGMENT REASSEMBLY QUEUE
 - ARP RECEIVE QUEUE
 - TCP REASSEMBLY QUEUE
 - SOCKET RECEIVE QUEUES
 - » **Send Queues:**
 - SOCKET SEND QUEUES
 - IP NETWORK INTERFACE SEND QUEUES

Network Traffic Analysis (Cont'd)



Ethereal Packet Analysis Timeline

Load vxWorks
T + 3.4 sec

Load startup.cmd
T + 23.5 sec

| No. | Time | Source | Destination | Protocol | Info |
|-------|------------|--------------|---------------|----------|--|
| 42 | 3.443301 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 44 | 3.444275 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000cat /ads/epicm/locCommon/scl-tpcf-loc05/vsWorks/vsWorks\000 |
| 1105 | 20.362914 | 172.31.72.79 | 172.31.72.101 | SSH | 1007\000 |
| 1114 | 20.364025 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 1115 | 20.364375 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000/bin/cat < /ads/epicm/locCommon/scl-tpcf-loc05/vsWorks/vsWorks.sys\000 |
| 1520 | 23.538910 | 172.31.72.79 | 172.31.72.101 | SSH | 1007\000 |
| 1527 | 23.541763 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 1529 | 23.542087 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000/bin/cat < /ads/epicm/locCommon/scl-tpcf-loc05/startup.cmd\000 |
| 1525 | 24.303799 | 172.31.72.79 | 172.31.72.101 | SSH | 1001\000 |
| 1532 | 24.310150 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 1534 | 24.310507 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000/bin/cat < /ads/epicm/locCommon/scl-tpcf-loc05/./all/common.cmd\000 |
| 1594 | 25.035115 | 172.31.72.79 | 172.31.72.101 | SSH | 999\000 |
| 1591 | 25.040553 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 1593 | 25.040905 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000/bin/cat < /ads/epicm/locCommon/scl-tpcf-loc05/vsWorks/vsWorks.cmd\000 |
| 50413 | 275.814750 | 172.31.72.79 | 172.31.72.101 | SSH | 1020\000 |
| 50420 | 275.820425 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 50423 | 275.821544 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000cat /ads/epicm/locCommon/scl-tpcf-loc05/vsWorks/vsWorks\000 |
| 50509 | 295.481831 | 172.31.72.79 | 172.31.72.101 | SSH | 1007\000 |
| 50517 | 295.485533 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 50519 | 295.485957 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000/bin/cat < /ads/epicm/locCommon/scl-tpcf-loc05/vsWorks/vsWorks.sys\000 |
| 50522 | 295.004881 | 172.31.72.79 | 172.31.72.101 | SSH | 1007\000 |
| 50531 | 295.005329 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 50533 | 295.005655 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000/bin/cat < /ads/epicm/locCommon/scl-tpcf-loc05/startup.cmd\000 |
| 51052 | 295.363038 | 172.31.72.79 | 172.31.72.101 | SSH | 1001\000 |
| 51059 | 295.367039 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 51071 | 295.368317 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000/bin/cat < /ads/epicm/locCommon/scl-tpcf-loc05/./all/common.cmd\000 |
| 51102 | 295.550797 | 172.31.72.79 | 172.31.72.101 | SSH | 999\000 |
| 51109 | 295.555175 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000 |
| 51111 | 295.555515 | 172.31.72.79 | 172.31.72.101 | SSH | vומר\000/bin/cat < /ads/epicm/locCommon/scl-tpcf-loc05/vsWorks/vsWorks.cmd\000 |

Frame 38 (50 bytes on wire (50 bytes captured))
Ethernet II, Src: 00:01:af:0e:25:19, Dst: 00:05:5b:3e:7b:3a
Internet Protocol, Src Addr: 172.31.72.79 (172.31.72.79), Dst Addr: 172.31.72.101 (172.31.72.101)
Transmission Control Protocol, Src Port: 1022 (1022), Dst Port: shell (514), Seq: 1, Ack: 1, Len: 5

Remote Shell (514), 5 bytes P: 54384 D: 33 M: 0

Network Traffic Analysis (Cont'd)



Ethereal Packet Analysis Timeline

vxWorks initialize
Restart NIC
T + 5.240 sec

NIC is ready again
T + 20.736 sec

EPICS neighbors
come knocking

| No. | Time | Source | Destination | Protocol | Info |
|------|-----------|-------------------|-------------------|------------------------|---|
| 1052 | 5.165047 | 172.31.74.121 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 1053 | 5.189988 | 172.31.72.82 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 1054 | 5.201289 | Motorola_04:bl:3c | Broadcast | ARP | Who has 172.31.72.53? Tell 172.31.72.16 |
| 1055 | 5.215079 | 172.31.73.154 | 172.31.75.255 | UDP | Source port: 5054 Destination port: 5055 |
| 1056 | 5.229296 | 172.31.73.41 | 255.255.255.255 | UDP | Source port: 44444 Destination port: 44444 |
| 1057 | 5.231960 | 172.31.72.57 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 1058 | 5.240273 | 172.31.74.90 | 172.31.75.255 | UDP | Source port: 1027 Destination port: 5055 |
| 1059 | 17.718368 | Cisco_d4:a8:41 | CDP/VTP | LLC | II, func=III; SAMP, OUI 0c00000c (Cisco), PID 0c2004 |
| 1070 | 17.719109 | Cisco_d4:a8:41 | CDP/VTP | LLC | II, func=III; SAMP, OUI 0c00000c (Cisco), PID 0c2004 |
| 1071 | 18.718028 | Cisco_d4:a8:41 | CDP/VTP | LLC | II, func=III; SAMP, OUI 0c00000c (Cisco), PID 0c2004 |
| 1072 | 18.719774 | Cisco_d4:a8:41 | CDP/VTP | LLC | II, func=III; SAMP, OUI 0c00000c (Cisco), PID 0c2004 |
| 1073 | 18.725003 | Cisco_d4:a8:41 | CDP/VTP | LLC | II, func=III; SAMP, OUI 0c00000c (Cisco), PID 0c2004 |
| 1074 | 18.725790 | Cisco_d4:a8:41 | CDP/VTP | LLC | II, func=III; SAMP, OUI 0c00000c (Cisco), PID 0c2004 |
| 1075 | 20.684963 | Motorola_0c:25:19 | Broadcast | ARP | Who has 172.31.72.78? Getantion ARP |
| 1076 | 20.736947 | Cisco_d4:a8:41 | | Spanning tree (for RRP | Success |
| 1077 | 20.741173 | 172.31.72.85 | 172.31.72.78 | TCP | 1077) 5054 [PSH, ACK] Seq=0 Ack=0 Win=8192 Len=32 TSV=2214038 TSEQ=94339 |
| 1078 | 20.741256 | 172.31.72.85 | 172.31.72.78 | TCP | 1078) 5054 [PSH, ACK] Seq=0 Ack=0 Win=8192 Len=32 TSV=2214038 TSEQ=94096 |
| 1079 | 20.741450 | Motorola_0c:25:19 | Broadcast | ARP | Who has 172.31.72.85? Tell 172.31.72.78 |
| 1080 | 20.741594 | Motorola_0b:7f:23 | Motorola_0c:25:19 | ARP | 172.31.72.85 in at 00:01:af:0b:7f:23 |
| 1081 | 20.741777 | 172.31.72.78 | 172.31.72.85 | TCP | 5054) 1078 [RST] Seq=0 Ack=3328835042 Win=0 Len=0 |
| 1082 | 20.746511 | 172.31.72.45 | 172.31.75.255 | UDP | Source port: 1033 Destination port: 5055 |
| 1083 | 20.746475 | 172.31.73.94 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 1084 | 20.755754 | 172.31.72.100 | 172.31.75.255 | UDP | Source port: 34040 Destination port: 5054 |
| 1085 | 20.776849 | 172.31.74.123 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 1086 | 20.789866 | 172.31.124.20 | 255.255.255.255 | UDP | Source port: 1032 Destination port: 5055 |
| 1087 | 20.832396 | 172.31.74.122 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 1088 | 20.832948 | 172.31.124.100 | 255.255.255.255 | UDP | Source port: 52988 Destination port: 5054 |
| 1089 | 20.833193 | 172.31.88.53 | 255.255.255.255 | UDP | Source port: 1029 Destination port: 5055 |
| 1090 | 20.833120 | 172.31.73.145 | 172.31.75.255 | UDP | Source port: 5054 Destination port: 5055 |
| 1091 | 20.835513 | 172.31.72.47 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |
| 1092 | 20.835501 | 172.31.72.105 | 172.31.75.255 | UDP | Source port: 1032 Destination port: 5055 |



Network Traffic Analysis (Cont'd)



Ethereal Packet Analysis Timeline

| No. | Time | Source | Destination | Protocol | Info |
|------|-----------|--------------|---------------|----------|--|
| 2130 | 30.730044 | 172.31.72.79 | 172.31.72.101 | NFS | V2 READ Call (Reply In 2133), DH:0x6f9320bc/bootLine |
| 2134 | 30.747532 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2135), DH:0x6f9320bc/bootLine |
| 2135 | 30.748070 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2137), DH:0x6f9320bc/ |
| 2138 | 30.748651 | 172.31.72.79 | 172.31.72.101 | NFS | V2 CREATE Call (Reply In 2139), DH:0x6f9320bc/bootLine |
| 2141 | 30.749499 | 172.31.72.79 | 172.31.72.101 | NFS | V2 READ Call (Reply In 2142), FH:0xb68b19ac Offset:0 Count:8192 TotalCount:0 |
| 2144 | 30.752058 | 172.31.72.79 | 172.31.72.101 | NFS | V2 WRITE Call (Reply In 2145), FH:0xb68b19ac BeginOffset:0 Offset:0 TotalCount:385 |
| 2317 | 32.827177 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2318), DH:0xcba1b652/linac |
| 2319 | 32.827772 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2320), DH:0x81904bc/8888 |
| 2321 | 32.828393 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2322), DH:0x81904bc/83-1-7 |
| 2323 | 32.828985 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2324), DH:0x84833ac/iscboot |
| 2325 | 32.829612 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2326), DH:0x84833ac/iscboot |
| 2327 | 32.830213 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2330), DH:0x70e44ac/st,cmd |
| 2331 | 32.840740 | 172.31.72.79 | 172.31.72.101 | NFS | V2 READ Call (Reply In 2333), FH:0x945e4dac Offset:0 Count:8192 TotalCount:0 |
| 2470 | 34.772333 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2471), DH:0xcba1b652/linac |
| 2472 | 34.773564 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2473), DH:0x81904bc/8888 |
| 2474 | 34.774172 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2475), DH:0x81904bc/83-1-7 |
| 2475 | 34.774881 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2477), DH:0x84833ac/iscboot |
| 2478 | 34.775454 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2479), DH:0x84833ac/iscboot |
| 2480 | 34.776038 | 172.31.72.79 | 172.31.72.101 | NFS | V2 LOOKUP Call (Reply In 2481), DH:0x70e44ac/st,cmd |

Frame 2129 (152 bytes on wire, 152 bytes captured)
 Ethernet II, Src: 00:01:af:0c:25:19, Dst: 00:06:5b:3e:7b:3a
 Internet Protocol, Src Addr: 172.31.72.79 (172.31.72.79), Dst Addr: 172.31.72.101 (172.31.72.101)
 User Datagram Protocol, Src Port: ircs (394), Dst Port: nfs (2049)
 Remote Procedure Call, Type:Call KID:0x19e837
 Network File System, LOOKUP Call DH:0x6f9320bc/bootLine

```

0000  00 06 5b 3e 7b 3a 00 01  af 0c 25 19 08 00 45 00  ..D.L...k...E.
0010  00 94 00 02 00 00 40 11  90 36 ac 1f 48 4e ac 1f  .....k...ll..
0020  48 55 03 e2 08 01 00 89  2e 44 01 9e 83 57 00 00  ..l...k...k...
0030  00 00 00 00 00 02 00 01  85 ad 00 00 00 02 00 00  .....k...k...
0040  00 04 00 00 00 01 00 00  00 24 00 00 27 10 00 00  .....k...k...
0050  00 0d 59 53 73 2d 51 53  53 5c 2d 73 72 75 31 00  ..l...ac...l...sv..
  
```

Do NFS
T + 30.7 sec

Network Traffic Analysis (Cont'd)



Ethereal Packet Analysis Timeline

RSTs from a previous connection

Do Etherlp
T + 87.16 sec

The screenshot shows a Wireshark packet capture window. The main pane displays a list of captured packets. The filter is set to 'ip.src == 172.31.72.78 && tcp.port == 44818'. The packet list includes several RST (Reset) packets from 172.31.72.78 to 172.31.74.50, followed by SYN and ACK packets to 172.31.74.49 and 172.31.72.90. The detailed view pane shows the structure of packet 1725: Ethernet II, Internet Protocol, and Transmission Control Protocol (Seq: 0, Ack: 1259438192, Len: 0). The packet bytes pane shows the raw hex and ASCII data.

| No. | Time | Source | Destination | Protocol | Info |
|-------|-----------|--------------|--------------|----------|--|
| 1725 | 25.206252 | 172.31.72.78 | 172.31.72.90 | TCP | 1027 > 44818 [RST] Seq=0 Ack=1259438192 Win=0 Len=0 |
| 2257 | 32.212903 | 172.31.72.78 | 172.31.74.50 | TCP | 1025 > 44818 [RST] Seq=0 Ack=84529559 Win=0 Len=0 |
| 2272 | 32.337578 | 172.31.72.78 | 172.31.74.49 | TCP | 1025 > 44818 [RST] Seq=0 Ack=88963037 Win=0 Len=0 |
| 11200 | 87.162368 | 172.31.72.78 | 172.31.74.49 | TCP | 1025 > 44818 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460 |
| 11201 | 87.162756 | 172.31.72.78 | 172.31.74.50 | TCP | 1025 > 44818 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460 |
| 11203 | 87.163224 | 172.31.72.78 | 172.31.72.90 | TCP | 1027 > 44818 [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1460 |
| 11205 | 87.163532 | 172.31.72.78 | 172.31.74.49 | TCP | 1025 > 44818 [ACK] Seq=1 Ack=1 Win=8192 Len=0 |
| 11206 | 87.163749 | 172.31.72.78 | 172.31.74.50 | TCP | 1025 > 44818 [ACK] Seq=1 Ack=1 Win=8192 Len=0 |
| 11207 | 87.164009 | 172.31.72.78 | 172.31.74.49 | ENMP | List Services (Req) |
| 11208 | 87.164283 | 172.31.72.78 | 172.31.74.50 | ENMP | List Services (Req) |
| 11210 | 87.165489 | 172.31.72.78 | 172.31.72.90 | TCP | 1027 > 44818 [ACK] Seq=1 Ack=1 Win=8192 Len=0 |
| 11212 | 87.165748 | 172.31.72.78 | 172.31.72.90 | ENMP | List Services (Req) |
| 11215 | 87.167251 | 172.31.72.78 | 172.31.74.50 | TCP | 1025 > 44818 [ACK] Seq=25 Ack=51 Win=8192 Len=0 |
| 11217 | 87.167534 | 172.31.72.78 | 172.31.74.50 | ENMP | Register Session (Req), Session: 0x00000000 |
| 11218 | 87.167757 | 172.31.72.78 | 172.31.74.49 | TCP | 1025 > 44818 [ACK] Seq=25 Ack=51 Win=8192 Len=0 |
| 11219 | 87.168040 | 172.31.72.78 | 172.31.74.49 | ENMP | Register Session (Req), Session: 0x00000000 |
| 11221 | 87.169121 | 172.31.72.78 | 172.31.74.50 | ENMP | Send RR Data (Req), Get Attribute Single |
| 11223 | 87.169617 | 172.31.72.78 | 172.31.74.49 | ENMP | Send RR Data (Req), Get Attribute Single |
| 11227 | 87.171554 | 172.31.72.78 | 172.31.72.90 | ENMP | Register Session (Req), Session: 0x00000000 |
| 11228 | 87.172033 | 172.31.72.78 | 172.31.74.50 | ENMP | Send RR Data (Req), Get Attribute Single |

Network Traffic Analysis (Cont'd)



Ethereal Packet Analysis Timeline

loclnit is running

T + 88.711 sec

Note:
EPICS ready after it sends
out its CA beacons

| No. | Time | Source | Destination | Protocol | Info |
|-------|-----------|--------------|----------------|----------|--|
| 11715 | 88.710365 | 172.31.72.78 | 172.31.75.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11717 | 88.711120 | 172.31.72.78 | 172.31.83.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11720 | 88.711497 | 172.31.72.78 | 172.31.91.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11721 | 88.711583 | 172.31.72.78 | 172.31.127.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11722 | 88.712107 | 172.31.72.78 | 172.31.75.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11723 | 88.712295 | 172.31.72.78 | 172.31.83.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11725 | 88.712530 | 172.31.72.78 | 172.31.91.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11727 | 88.713197 | 172.31.72.78 | 172.31.127.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11738 | 88.745401 | 172.31.72.78 | 172.31.75.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11739 | 88.745680 | 172.31.72.78 | 172.31.83.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11740 | 88.745794 | 172.31.72.78 | 172.31.91.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11741 | 88.745917 | 172.31.72.78 | 172.31.127.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11755 | 88.812448 | 172.31.72.78 | 172.31.75.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11756 | 88.812525 | 172.31.72.78 | 172.31.83.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11759 | 88.813035 | 172.31.72.78 | 172.31.91.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11770 | 88.813209 | 172.31.72.78 | 172.31.127.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11829 | 88.998428 | 172.31.72.78 | 172.31.75.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11830 | 88.998577 | 172.31.72.78 | 172.31.83.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11831 | 88.998738 | 172.31.72.78 | 172.31.91.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11832 | 88.998899 | 172.31.72.78 | 172.31.127.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11955 | 89.312511 | 172.31.72.78 | 172.31.75.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11956 | 89.312543 | 172.31.72.78 | 172.31.83.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11957 | 89.312845 | 172.31.72.78 | 172.31.91.255 | UDP | Source port: 1031 Destination port: 5065 |
| 11958 | 89.313009 | 172.31.72.78 | 172.31.127.255 | UDP | Source port: 1031 Destination port: 5065 |
| 12141 | 89.945698 | 172.31.72.78 | 172.31.75.255 | UDP | Source port: 1031 Destination port: 5065 |
| 12142 | 89.945759 | 172.31.72.78 | 172.31.83.255 | UDP | Source port: 1031 Destination port: 5065 |
| 12143 | 89.945933 | 172.31.72.78 | 172.31.91.255 | UDP | Source port: 1031 Destination port: 5065 |
| 12144 | 89.946097 | 172.31.72.78 | 172.31.127.255 | UDP | Source port: 1031 Destination port: 5065 |
| 19529 | 91.926076 | 172.31.72.78 | 172.31.75.255 | UDP | Source port: 1031 Destination port: 5065 |

| | |
|--|---|
| Frame 11715 (50 bytes on wire (50 bytes captured) on interface eth0): | |
| Ethernet II, Src: 00:01:af:0c:25:19, Dest: ff:ff:ff:ff:ff:ff | |
| Destination: ff:ff:ff:ff:ff:ff (Broadcast) | |
| Source: 00:01:af:0c:25:19 (Motorola_0c:25:19) | |
| Type: IP (0x0800) | |
| Trailer: 0000 | |
| Internet Protocol, Src Addr: 172.31.72.78 (172.31.72.78), Dest Addr: 172.31.75.255 (172.31.75.255) | |
| User Datagram Protocol, Src Port: 1031 (1031), Dest Port: 5065 (5065) | |
| Source port: 1031 (1031) | |
| Destination port: 5065 (5065) | |
| Length: 24 | |
| Checksum: 0xf313 (correct) | |
| Data (15 bytes) | |
| 0000 | ff ff ff ff ff ff 00 01 af 0c 25 19 08 00 45 00 |
| 0010 | 00 2c 05 ec 00 00 40 11 88 49 ac 1f 48 4e ac 1f |
| 0020 | 4b ff 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| 0030 | 13 c8 00 00 00 00 ac 1f 48 4e 00 00 |

Network Traffic Analysis (Cont'd)



Ethereal Packet Analysis Timeline

RSTs from a previous connection

Talk to Archiver
T + 90.73 sec

| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|-----------|--------------|----------------|----------|-------------------|--|
| 12511 | 90.707339 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 50945 [RST] Seq=0 Ack=1471157515 Win=0 Len=0 |
| 12513 | 90.730142 | 172.31.72.79 | 172.31.124.100 | HTTP | Source port: 5094 | Destination port: 50945 |
| 12519 | 90.731205 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=0 TSV=129 TSER=1135040140 |
| 12530 | 90.737273 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=1 Ack=1081 Win=8192 Len=0 TSV=129 TSER=1135040140 |
| 12531 | 90.738540 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [PSH, ACK] Seq=1 Ack=1081 Win=8192 Len=540 TSV=129 TSER=1135040140 |
| 12537 | 90.740547 | 172.31.72.79 | 172.31.124.100 | HTTP | Source port: 5094 | Destination port: 50945 |
| 12540 | 90.752187 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=541 Ack=2473 Win=8192 Len=0 TSV=129 TSER=1135040143 |
| 12543 | 90.800561 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=541 Ack=2557 Win=8192 Len=0 TSV=129 TSER=1135040143 |
| 12554 | 90.804351 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=541 Ack=2557 Win=8192 Len=1448 TSV=129 TSER=1135040143 |
| 12555 | 90.804633 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=2088 Ack=2557 Win=8192 Len=1448 TSV=129 TSER=1135040143 |
| 12555 | 90.804549 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [PSH, ACK] Seq=3537 Ack=2557 Win=8192 Len=948 TSV=129 TSER=1135040143 |
| 12568 | 90.806405 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [PSH, ACK] Seq=4385 Ack=2557 Win=8192 Len=948 TSV=129 TSER=1135040147 |
| 12571 | 90.806035 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [PSH, ACK] Seq=5033 Ack=3573 Win=8192 Len=1088 TSV=129 TSER=1135040147 |
| 12573 | 90.806513 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [PSH, ACK] Seq=6121 Ack=3573 Win=8192 Len=552 TSV=129 TSER=1135040147 |
| 12574 | 90.806018 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [PSH, ACK] Seq=6473 Ack=3573 Win=8192 Len=102 TSV=129 TSER=1135040148 |
| 12710 | 91.300352 | 172.31.72.79 | 172.31.124.100 | HTTP | Source port: 5094 | Destination port: 45292 |
| 12712 | 91.301232 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=0 TSV=130 TSER=1135040197 |
| 12717 | 91.303933 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=1 Ack=1081 Win=8192 Len=0 TSV=130 TSER=1135040197 |
| 12718 | 91.306475 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [PSH, ACK] Seq=1 Ack=1081 Win=8192 Len=758 TSV=130 TSER=1135040197 |
| 12722 | 91.306354 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=759 Ack=1729 Win=8192 Len=0 TSV=130 TSER=1135040197 |
| 12723 | 91.308142 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=759 Ack=1729 Win=8192 Len=1448 TSV=130 TSER=1135040197 |
| 12724 | 91.308253 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=2217 Ack=1729 Win=8192 Len=1448 TSV=130 TSER=1135040197 |
| 12725 | 91.308394 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [PSH, ACK] Seq=3555 Ack=1729 Win=8192 Len=1200 TSV=130 TSER=1135040197 |
| 12726 | 91.308055 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [PSH, ACK] Seq=4855 Ack=1729 Win=8192 Len=344 TSV=130 TSER=1135040198 |
| 12732 | 91.310431 | 172.31.72.79 | 172.31.124.100 | HTTP | Source port: 5094 | Destination port: 45292 |
| 12733 | 91.311284 | 172.31.72.79 | 172.31.124.100 | HTTP | Source port: 5094 | Destination port: 45292 |
| 12735 | 91.311280 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=5009 Ack=1889 Win=8192 Len=0 TSV=130 TSER=1135040198 |
| 12736 | 91.313025 | 172.31.72.79 | 172.31.124.100 | TCP | 5094 | 5094 > 32828 [ACK] Seq=5009 Ack=1889 Win=8192 Len=0 TSV=130 TSER=1135040198 |

1 - Frame 2537 (50 bytes on wire (50 bytes captured))
 - Ethernet II, Src: 00:01:af:0c:25:19, Dst: 00:00:00:07:ac:0a
 Destination: 00:00:00:07:ac:0a (All-MSRP-routers-0a)
 Source: 00:01:af:0c:25:19 (Motorola_0c:25:19)
 Type: IP (0x0800)
 Trailer: 000000000000
 - Internet Protocol, Src Addr: 172.31.72.79 (172.31.72.79), Dst Addr: 172.31.124.100 (172.31.124.100)
 - Transmission Control Protocol, Src Port: 5094 (5094), Dst Port: 50945 (50945), Seq: 0, Ack: 108003345, Len: 0

0000 00 00 00 07 ac 0a 01 af 0c 25 19 00 00 45 00
 0010 00 21 00 f7 00 00 40 00 5c 03 ac 1f 40 4e ac 1f
 0020 7c 04 13 c8 ca 0c f2 0a 1c 85 00 00 00 50 04
 0030 00 00 35 a7 00 00 00 00 00 00 00 00

