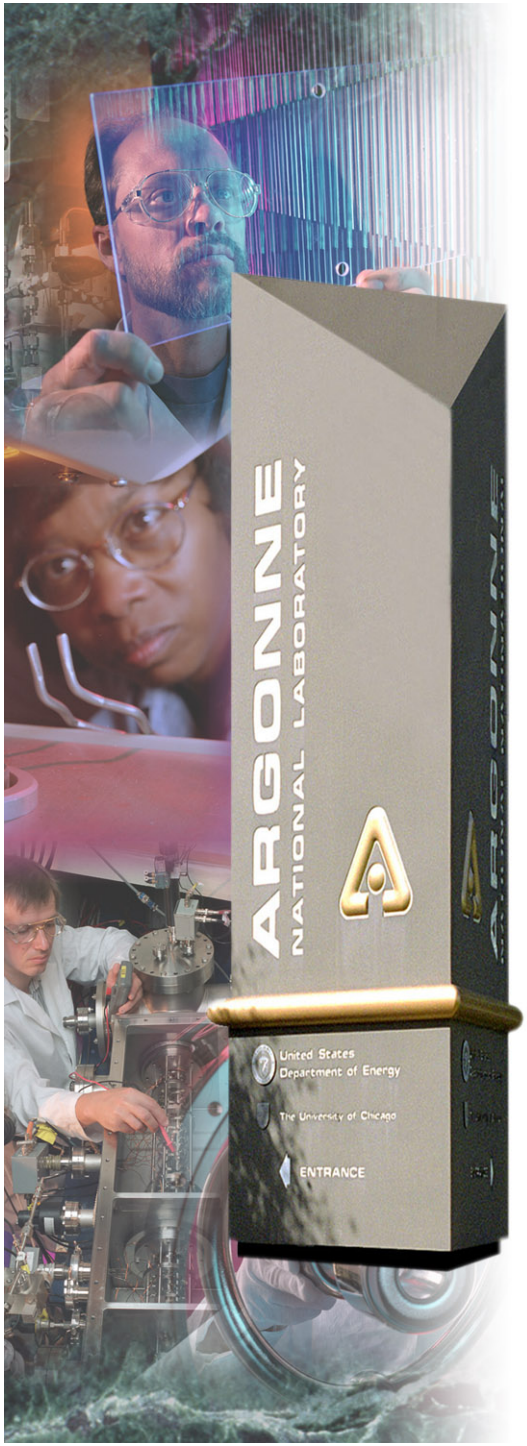



# *Channel Access in Depth*

*Kenneth Evans, Jr.*

*March 8, 2005*

*Part of the EPICS "Getting Started" Lecture Series*

***Argonne National Laboratory***

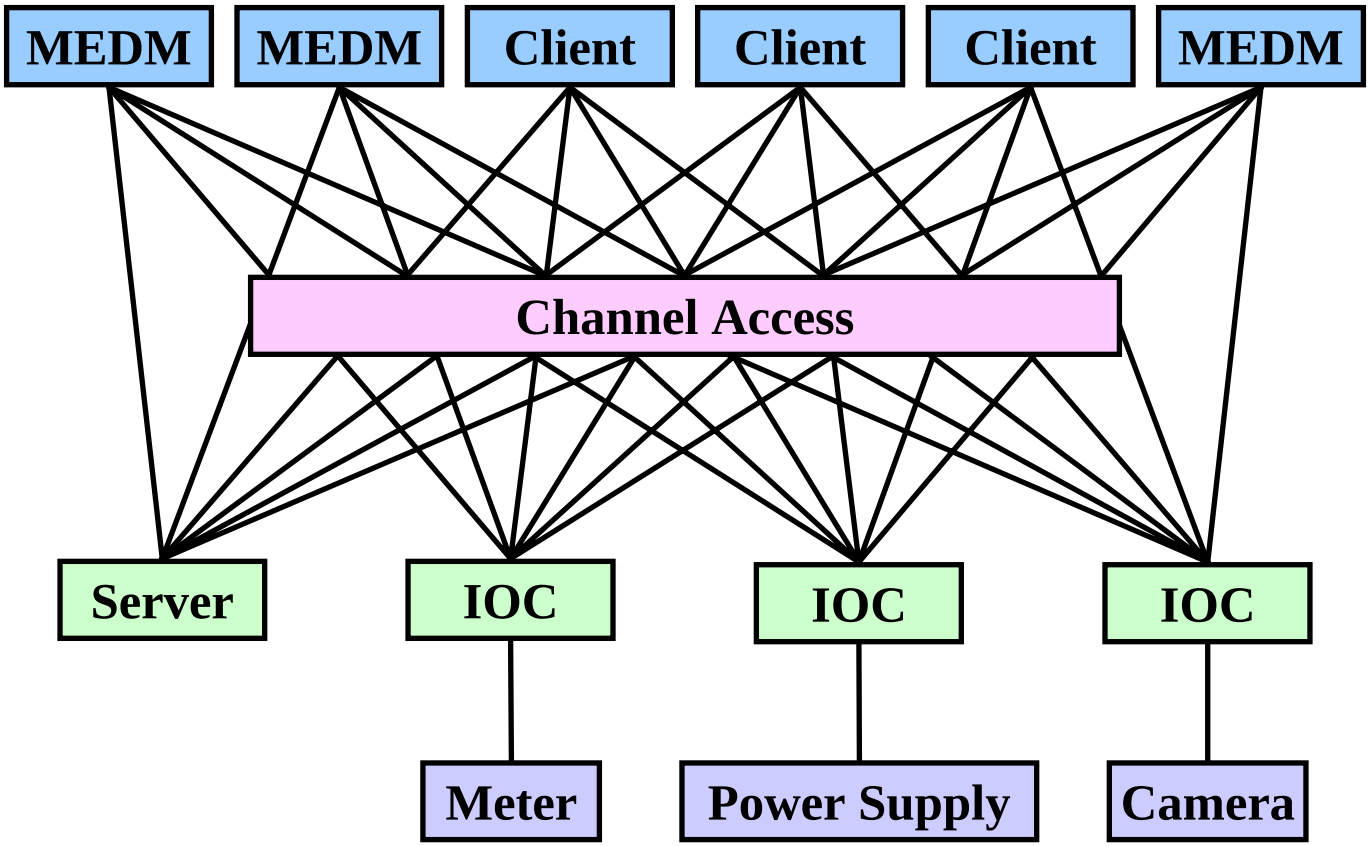
Office of Science
U.S. Department of Energy

*A U.S. Department of Energy*
*Office of Science Laboratory*
*Operated by The University of Chicago*

# *Topics Covered*

- **Channel Access**
  - The means by which EPICS Clients and Servers talk to each other
- **CaSnooper**
  - An application to monitor Search Requests for Process variables
- **CASW**
  - An application to monitor Beacon Anomalies
- **ParseCASW**
  - An application to convert CASW output to identifiable events
- **RunCaSnooper**
  - An application that provides an interface to CaSnooper and CASW with an associated MEDM and StripTool

# EPICS Overview



# *Channel Access Concepts*

- **Network Protocols**
- **Process Variable Connection Process**
- **Search Request**
- **Exist Test**
- **Beacons**
- **Beacon Anomaly**
- **CaRepeater**

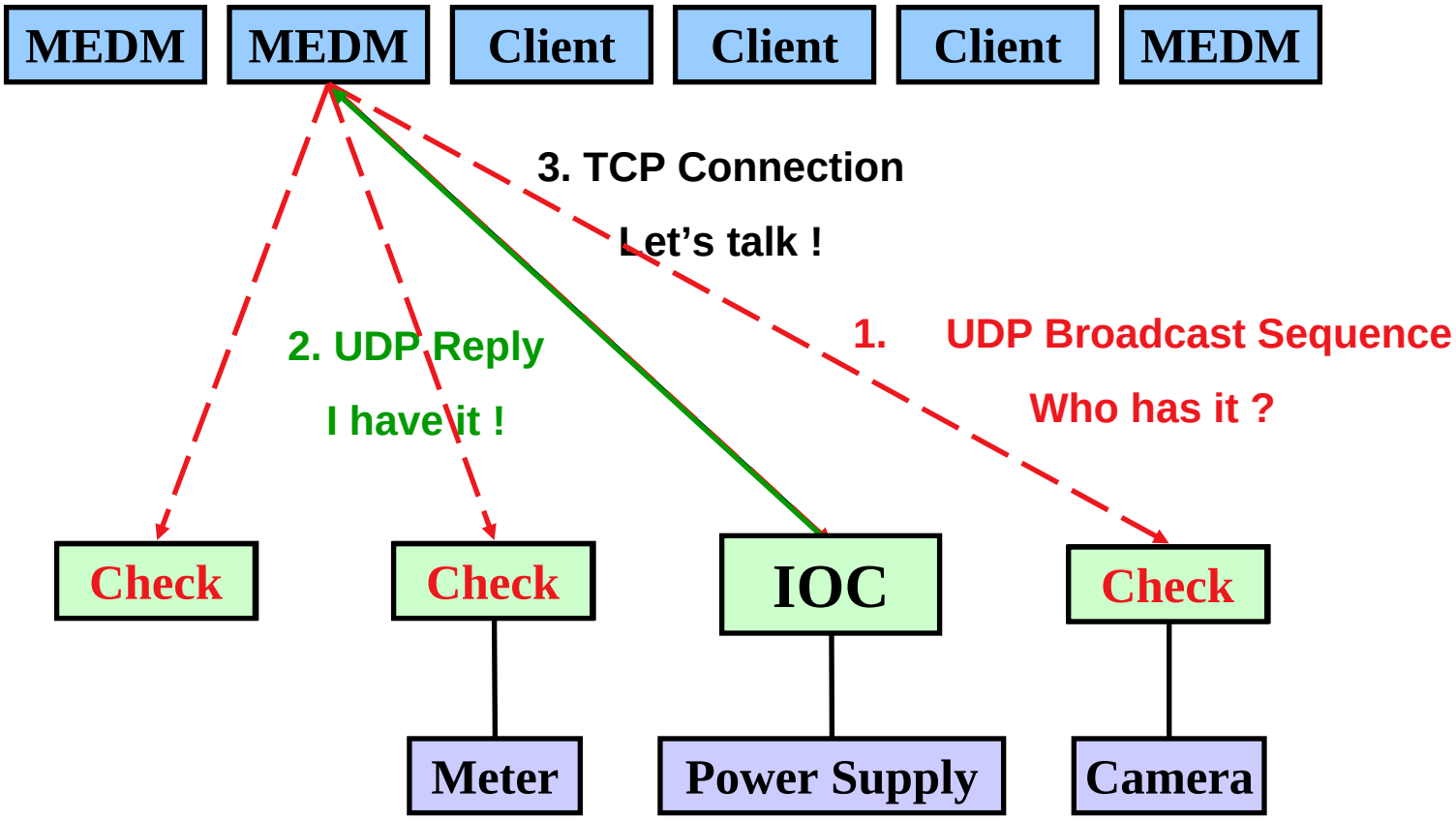
# *Network Protocols*

- **Channel Access uses two Network Protocols, UDP and TCP**
- **UDP (User Datagram Protocol)**
  - One way, unreliable
  - Send out packets, no guarantee they reach their destination
  - Can be broadcast or directed (unicasts)
    - *Broadcasts: To all IP addresses, e.g. 123.45.6.255*
    - *Unicasts: To a specific IP address, e.g. 123.45.6.100*
  - Broadcasts may not leave subnets for security reasons
- **TCP (Transmission Control Protocol)**
  - Two way, reliable, persistent
  - Socket at each end
  - Acknowledgements, timeouts, retransmissions, etc. guarantee reliability

# *Connection Process*

- **A client (e.g. MEDM) wanting a PV sends a UDP search request**
  - Sent to EPICS_CA_ADDR_LIST
  - (Or its default -- broadcast to all interfaces on the host machine)
  - Sent on EPICS_CA_SERVER_PORT [5064]
  - Do you have this PV?
- **Each Server that gets a packet does an exist test**
  - Do I have this PV?
- **Server with the PV sends a directed UDP reply to the Client**
  - I have this PV.
- **A TCP connection is established between the Server and the Client (or an existing one is used)**
  - One per Client-Server pair, no matter how many PVs
  - Referred to as a Circuit
  - Let's talk.

# Search and Connect Graphically



# *Search Request*

- **A client makes a search request when it wants to find out what server has the PV**
  - Happens when a PV is first created in the client
  - On a beacon anomaly (unresolved PVs only)
  - When another PV is created (unresolved PVs only)
- **A search request consists of a sequence of UDP packets**
  - Starts with a small interval (30 ms), that doubles each time
  - Until it gets larger than 5 s, then it stays at 5 s

  - Stops after 100 packets or when it gets a response
  - Used to never try again until it sees a beacon anomaly or creates a new PV
    - *As of 3.14.7 retries at a slow rate*
  - Total time is about 8 minutes to do all 100
  - The sequence may be different owing to fine tuning
- **Usually connects on the first packet or the first few**

# *Exist Test*

- **Every time a Server receives a search request packet, its pvExistTest routine is called**
- **The Server has to check if it has the PV**
  - Returns ExistsHere or DoesNotExistHere
- **Normally a search request sequence ends after a few packets**
  - Because one Server soon returns ExistsHere
- **For PVs that do not exist**
  - There are 100 tests per search request sequence for that PV
  - This happens every time a Client initiates a search request sequence
    - *Each time the Client searches for a new PV*
    - *At each beacon anomaly, perceived or real*

# Beacons

- **A Beacon is a UDP broadcast packet sent by a Server**
- **When it is healthy, each Server broadcasts a UDP beacon at regular intervals (like a heartbeat)**
  - EPICS_CA_BEACON_PERIOD, 15 s by default

- **When it is coming up, each Server broadcasts a startup sequence of UDP beacons**
  - Starts with a small interval (25 ms, 75 ms for VxWorks)
  - Interval doubles each time
  - Until it gets larger than 15 s, then it stays at 15 s

  - Takes about 10 beacons and 40 s to get to steady state
- **Clients monitor the beacons**
  - Determine connection status, whether to reissue searches

# *Beacon Anomaly*

- **A Beacon Anomaly is any change from the normal beacon interval (15 s)**
- **No beacons:**
  - After 30 sec the client sends message over TCP connection
  - If no beacons and no reply, connection is down
  - That is when MEDM screens go white
- **Abnormal interval:**
  - Short: IOC has come up
  - Long: IOC was disconnected
- **May cause clients to reissue outstanding search requests**
- **Network problems can look like beacon anomalies**

# *Virtual Circuit Disconnect*

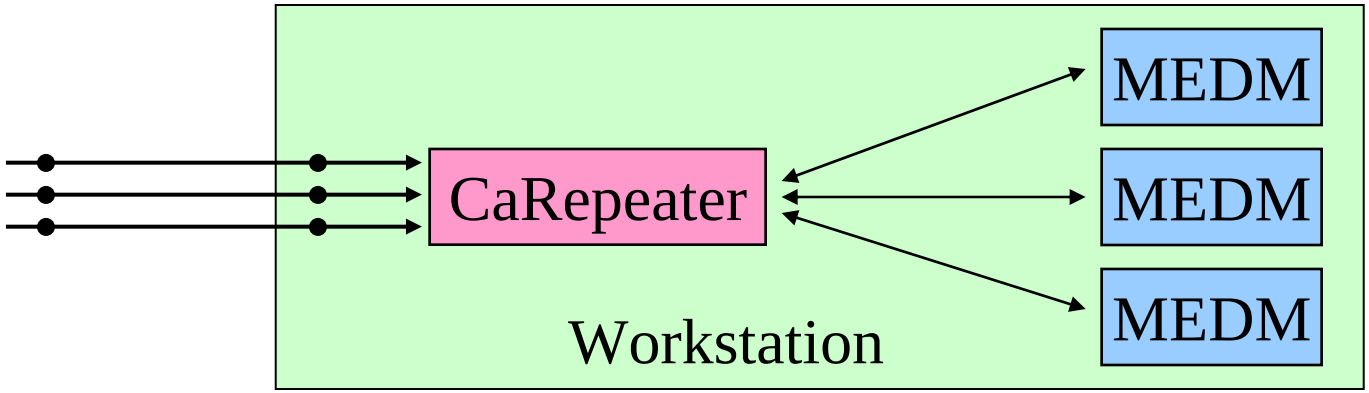
- **3.13 and early 3.14**
  - Hang-up message or no response from server for 30 sec.
  - If not a hang-up, then client sends “Are you there” query
  - If no response for 5 sec, TCP connection is closed
  - MEDM screens go white
  - Clients reissue search requests
- **3.14.5 and later**
  - Hang-up message from server
  - TCP connection is closed
  - MEDM screens go white
  - Clients reissue search requests

# *Virtual Circuit Unresponsive*

- **3.14.5 and later**
  - No response from server for 30 sec.
  - Client then sends “Are you there” query
  - If no response for 5 sec, TCP connection is not closed
    - *For several hours, at least*
  - MEDM screens go white
  - Clients do not reissue search requests
    - *Helps with network storms*
  - Clients that do not call ca_poll frequently get a virtual circuit disconnect even though the server may be OK
    - *Clients written for 3.13 but using 3.14 may have a problem*
    - *May be changed in future versions*

# *CaRepeater*

- **UDP broadcasts are not guaranteed to go to every process on a workstation**
- **CaRepeater solves this problem**
  - There is one CaRepeater process per workstation
  - Clients make a TCP connection to it when they start up
  - CaRepeater receives the beacons
    - *EPICS_CA_REPEATER_PORT [usually 5065]*
  - CaRepeater forwards the beacons to the Client
- **This problem does not exist on most modern systems**



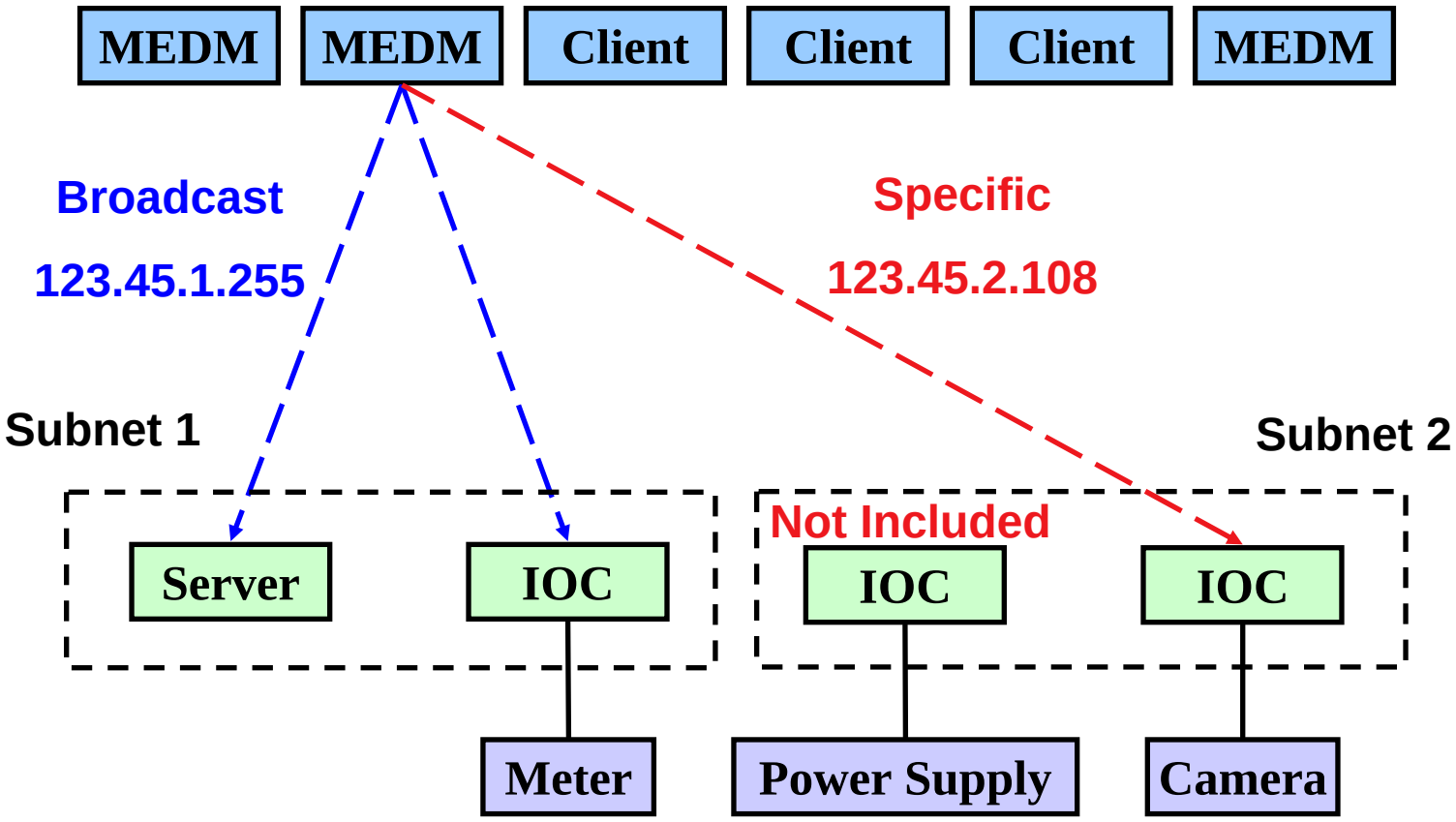
# *Multiple Servers on the Same Host*

- **Used to not be possible at all (Base 3.13)**
- **Now, it can be done, but there are problems**
- **Will get message**
  - cas warning: Configured TCP port was unavailable. Using dynamically assigned TCP port 45003, but now two or more servers share the same UDP port. Depending on your IP kernel this server may not be reachable with UDP unicast (a host's IP in EPICS_CA_ADDR_LIST)
- **First part means clients will establish their circuit on another port than the default 5064.**
  - Not a problem
- **Second part means unicast search requests may not get to both servers**
  - UDP deficiency, similar to the CaRepeater problem
  - May be a problem when EPICS_CA_ADDR_LIST is used

# *Important Environment Variables*

- **EPICS_CA_ADDR_LIST**
  - Determines where to search
  - Is a list (separated by spaces)
    - *"123.45.1.255 123.45.2.14 123.45.2.108"*
  - Default is broadcast addresses of all interfaces on the host
    - *Works when servers are on same subnet as clients*
  - Broadcast address
    - *Goes to all servers on a subnet*
    - *Example: 123.45.1.255*
    - *Use ifconfig –a on UNIX to find it (or ask an administrator)*
- **EPICS_CA_AUTO_ADDR_LIST**
  - YES: Include default addresses above in searches
  - NO: Do not search on default addresses
  - If you set EPICS_CA_ADDR_LIST, usually set this to NO

# EPICS_CA_ADDR_LIST



# *Other Environment Variables*

- **CA Client**

**EPICS_CA_ADDR_LIST**
**EPICS_CA_AUTO_ADDR_LIST**
**EPICS_CA_CONN_TMO**
**EPICS_CA_BEACON_PERIOD**
**EPICS_CA_REPEATER_PORT**
**EPICS_CA_SERVER_PORT**
**EPICS_CA_MAX_ARRAY_BYTES**
**EPICS_TS_MIN_WEST**

- **CA Server**

**EPICS_CAS_SERVER_PORT**
**EPICS_CAS_AUTO_BEACON_ADDR_LIST**
**EPICS_CAS_BEACON_ADDR_LIST**
**EPICS_CAS_BEACON_PERIOD**
**EPICS_CAS_BEACON_PORT**
**EPICS_CAS_INTF_ADDR_LIST**
**EPICS_CAS_IGNORE_ADDR_LIST**

- **See the Channel Access Reference Manual for more information**

# Summary

- **Clients send search requests when they want a PV**
- **Each server has to check if it has the PV for every packet in the search-request sequence**
- **Servers send beacons at regular intervals and with a faster pattern when they come up**
- **A beacon anomaly is any pattern that is not a regular beacon**
- **Beacon anomalies may cause clients to resend search requests for any unresolved PVs**
- **Search request sequences end early for found PVs but not for non-existent PVs**
- **Search requests put a load on the servers and add to network traffic**
  - This can cause problems
  - Consequently, undesirable beacon anomalies and search requests should be minimized or eliminated
- **Searches are on port 5064 and beacons are on port 5065**

# *CaSnooper*

- **CaSnooper is a server whose ExistTest routine keeps track of search requests rather than seeing if it has the PV**
- **It can print the names of all PVs being searched for and related statistics using several report formats**
- **It can also check if these PVs are connected (C) or not (NC)**
- **It has internal PVs if started with the –n option**
  - ExistTest rates that can be monitored
  - Others that allow it to be controlled from an MEDM screen
  - The PV prefix [default CaSnoop] can be changed to prevent collisions
- **Running CaSnooper:**
  - Run at the command line to get one report
  - Run with PVs for monitoring, say with SDDSmonitor or StripTool
  - Run with PVs and control with MEDM for continuous operation
- **To run CaSnooper you may need the full path**
  - /usr/local/epics/extensions/bin/solaris-sparc/caSnooper
  - It is not installed at the APS for 3.13 (3.13 servers have less capability)

# Sample CaSnooper Output

Two lines from RunCaSnooper →

Print top 10 (-p10) →

Check top 10 (-c10) →

CaSnooper

```
Starting CcSnooper...
Type Ctrl-C to stop it
Starting CaSnooper 2.1.0.1 (8-27-2003) at Nov 04 13:50:08
EPICS 3.14.3
Individual Name is CaSnoop.test
PV name prefix is CaSnoop

Nov 04 13:50:42:
There were 17508 requests to check for PV existence for 2307 different PVs.
  Max(Hz):   11.17
  Mean(Hz):  0.22
  StDev(Hz): 0.61

PVs with top 10 requests:
   1 willow:52275                FEL:$(M8).VAL                 11.17
   2 willow:52275                FEL:$(M8).RBV                 8.94
   3 gateway433:33790            FE:09:ID:SR:HPOS:CC.VAL       4.15
   4 gateway433:33790            FE:09:ID:SR:VPOS:CC.VAL       4.15
   5 willow:52275                FEL:VUV4f2_able.VAL           3.35
   6 gateway435:33270            s17id:scan1.NPTS              3.06
   7 gateway435:33270            s17id:scan1.MPTS              3.06
   8 gateway435:33270            s17id:scan1.PASM              3.06
   9 gateway435:33270            s17id:scan1.P1EP              3.06
  10 gateway435:33270            s17id:scan1.P1AR              3.06

Nov 04 13:51:01:
There were 23868 requests to check for PV existence for 2329 different PVs.
  Max(Hz):   10.45
  Mean(Hz):  0.19
  StDev(Hz): 0.56

Connection status for top 10 PVs after 10.00 sec:
   1 willow:52275                FEL:$(M8).VAL                 NC 10.45
   2 willow:52275                FEL:$(M8).RBV                 NC 8.36
   3 gateway433:33790            FE:09:ID:SR:HPOS:CC.VAL       NC 4.01
   4 gateway433:33790            FE:09:ID:SR:VPOS:CC.VAL       NC 4.01
   5 willow:52275                FEL:VUV4f2_able.VAL           NC 3.13
   6 gateway435:33270            s17id:scan1.NPTS              NC 3.04
   7 gateway435:33270            s17id:scan1.P1PV              NC 3.04
   8 gateway435:33270            s17id:scan1.R1PV              NC 3.04
   9 gateway435:33270            s17id:scan1.CMND              NC 3.04
  10 gateway435:33270            s17id:scan1.P1EP              NC 3.04
```

individual name, prefix

statistics

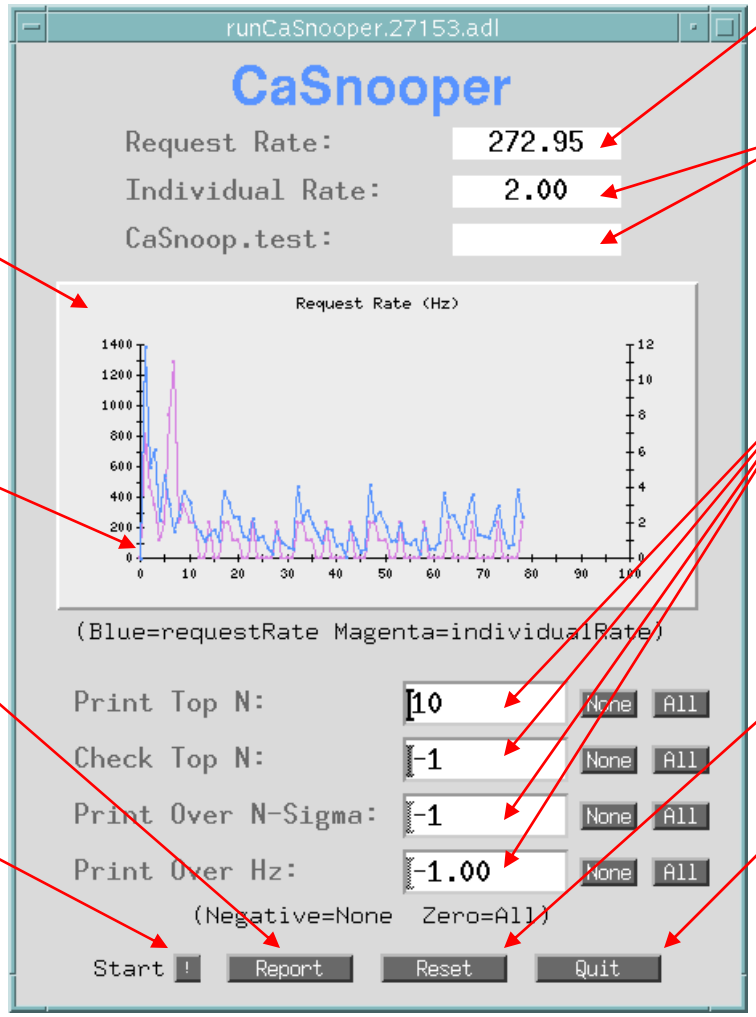
machine:port, (can be used to identify source)

name

search rate in Hz

Not connected, will be C for connected (hardly ever the case)

# Control CaSnooper via MEDM




Request rate

Individual rate for CaSnoop.test, which doesn't exist

Cartesian plot of requestRate and individualRate

CaSnooper was started here (with EPICS_CA_REPEATER_PORT = 5065)

Use these to set what will happen when you press Report. Case illustrated will print the top 10.

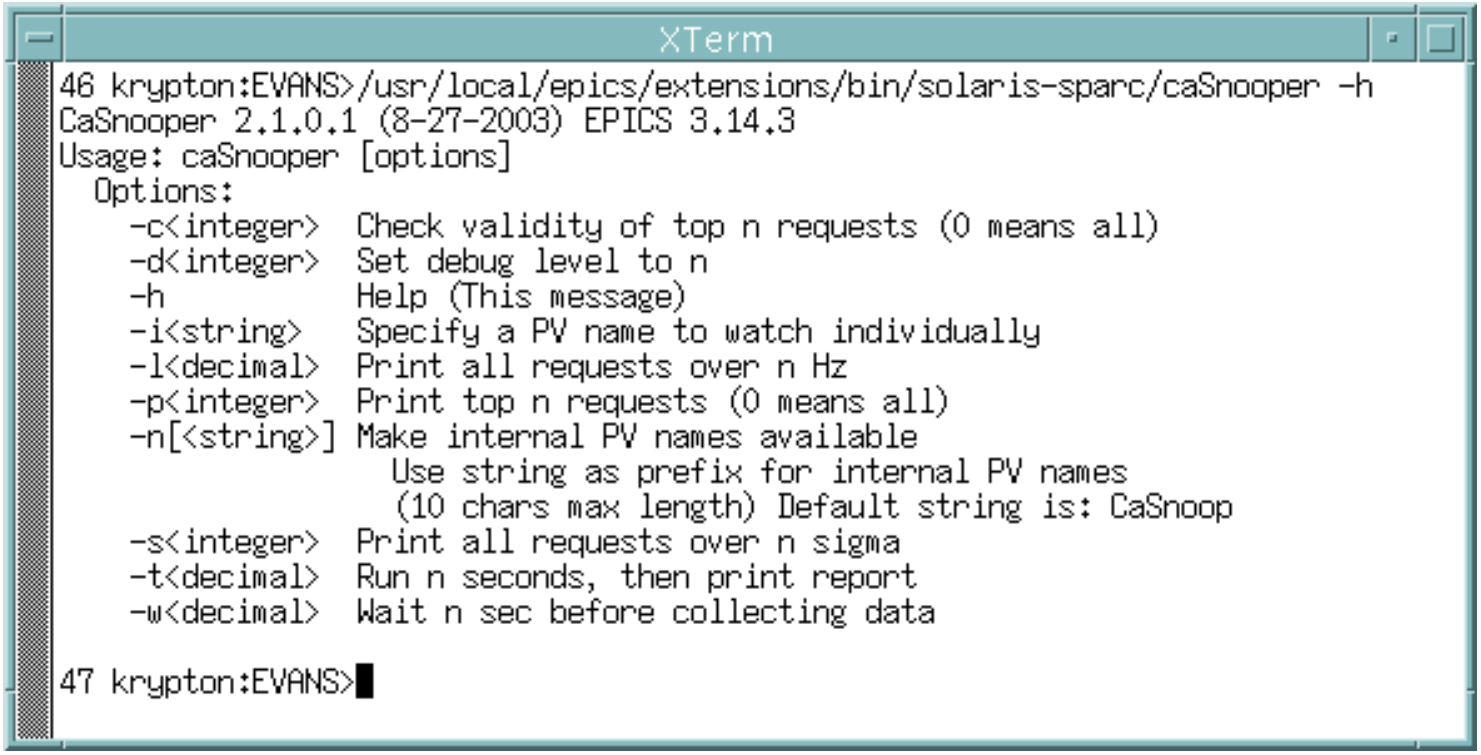
Execute selected reports in the CaSnooper stdout

Reset the counters in CaSnooper

Shell command to start CaSnooper, CASW, StripTool, etc.

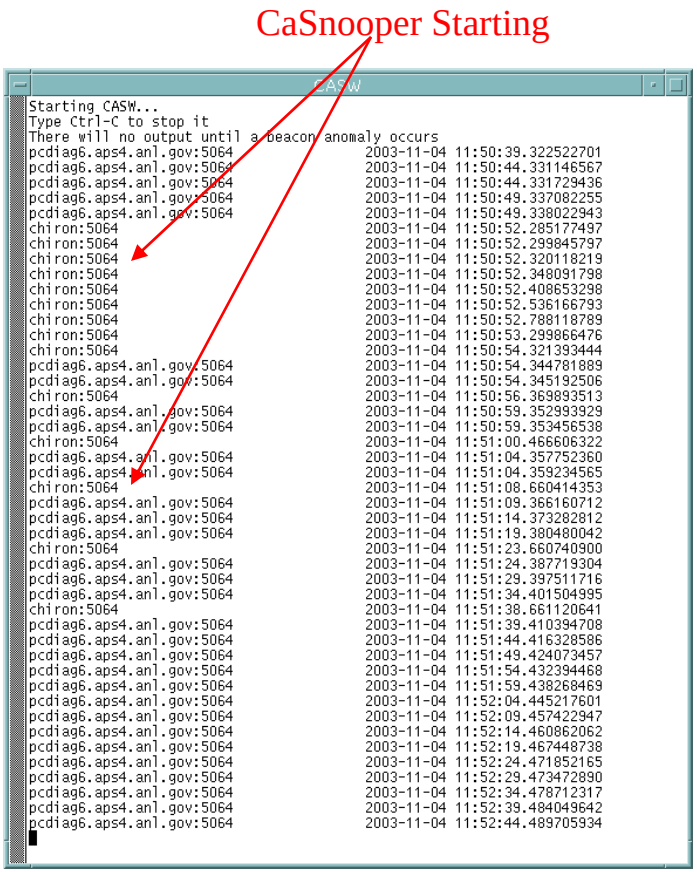
Stop CaSnooper

# CaSnooper Options



# CASW

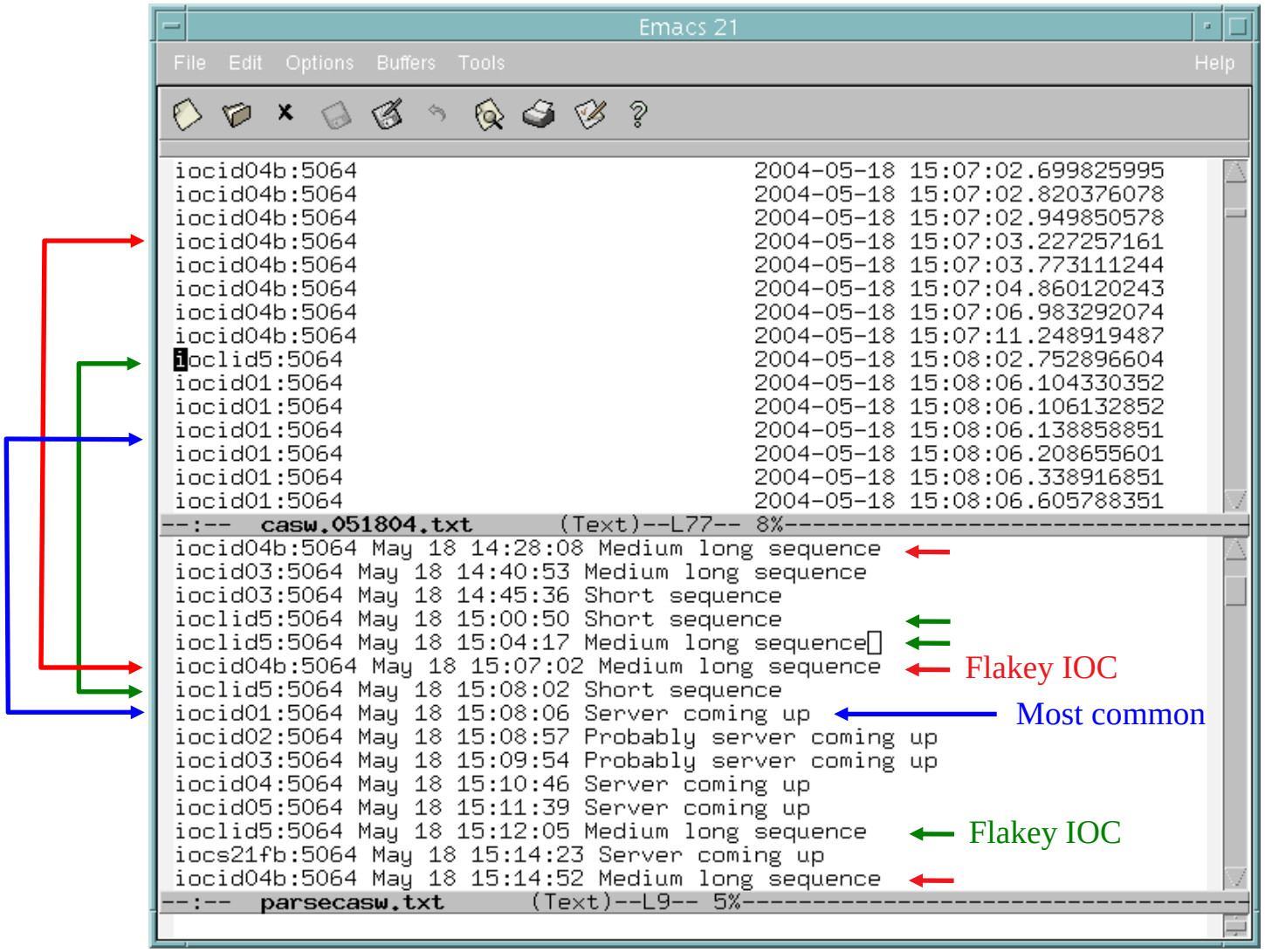
- **CASW (Channel Access Server Watcher) monitors Beacon Anomalies**
- **Is a simple command-line utility**
- **Part of EPICS Base**
- **May need a full path to the version of base desired**
  - 3.14 (Recommended) /usr/local/epics/base3.14.3/bin/solaris-sparc/casw
  - 3.13: /usr/local/epics/base/bin/solaris/casw
- **Prints a line with a timestamp when it sees a beacon anomaly**



# *ParseCASW*

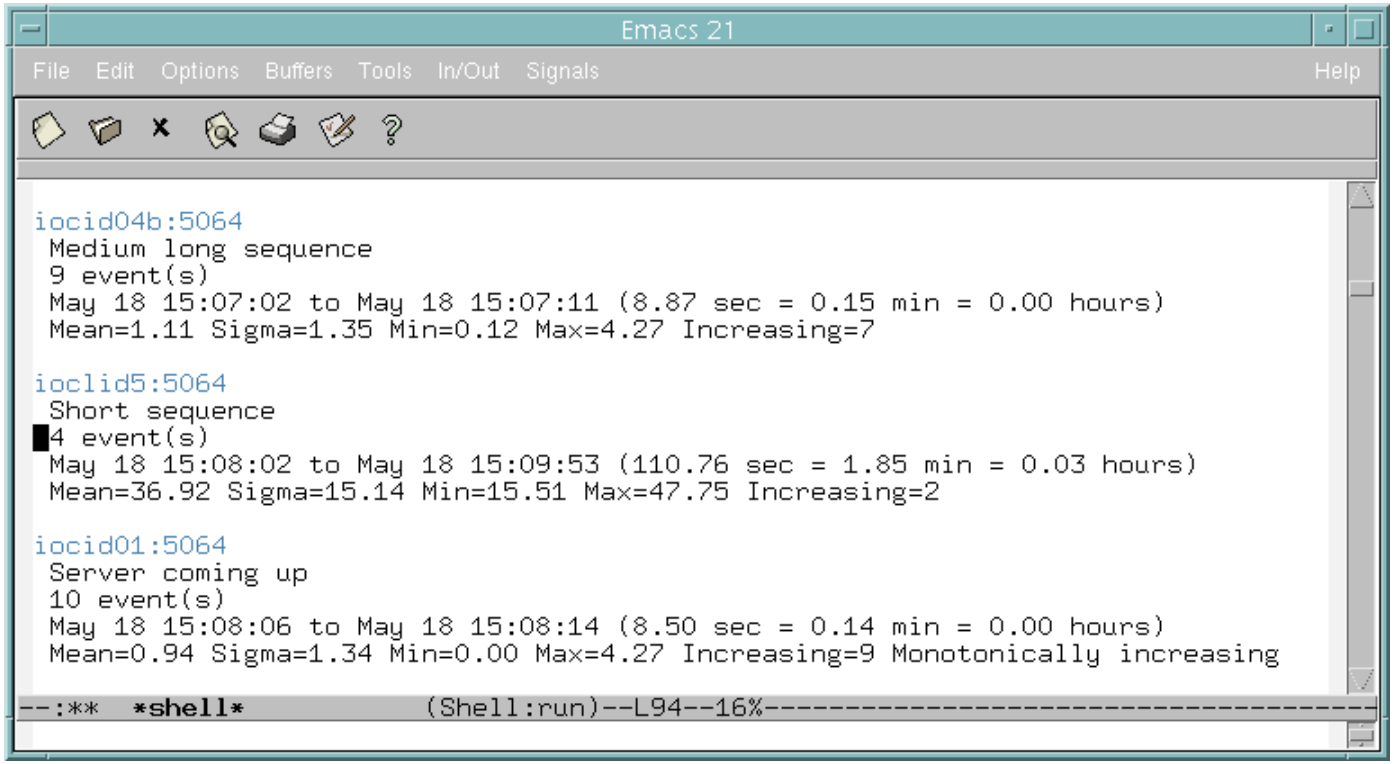
- **CASW produces a list of beacons that came at the wrong time**
  - Listed in the order they happen
    - *Intervals between anomalies are important*
    - *Not easy to see from the output*
  - Anomalies from different causes are mixed together
  - Cause of an anomaly sequence is not readily apparent
- **ParseCASW parses the CASW output**
  - Uses artificial intelligence to try to determine the event that caused the anomaly (e.g. an IOC coming up)
  - Prints the events instead of the anomalies
- **Can be used in two ways**
  - Parse output saved from CASW (or OAG data logging)
  - Pipe CASW into ParseCASW in real time

# ParseCASW Output

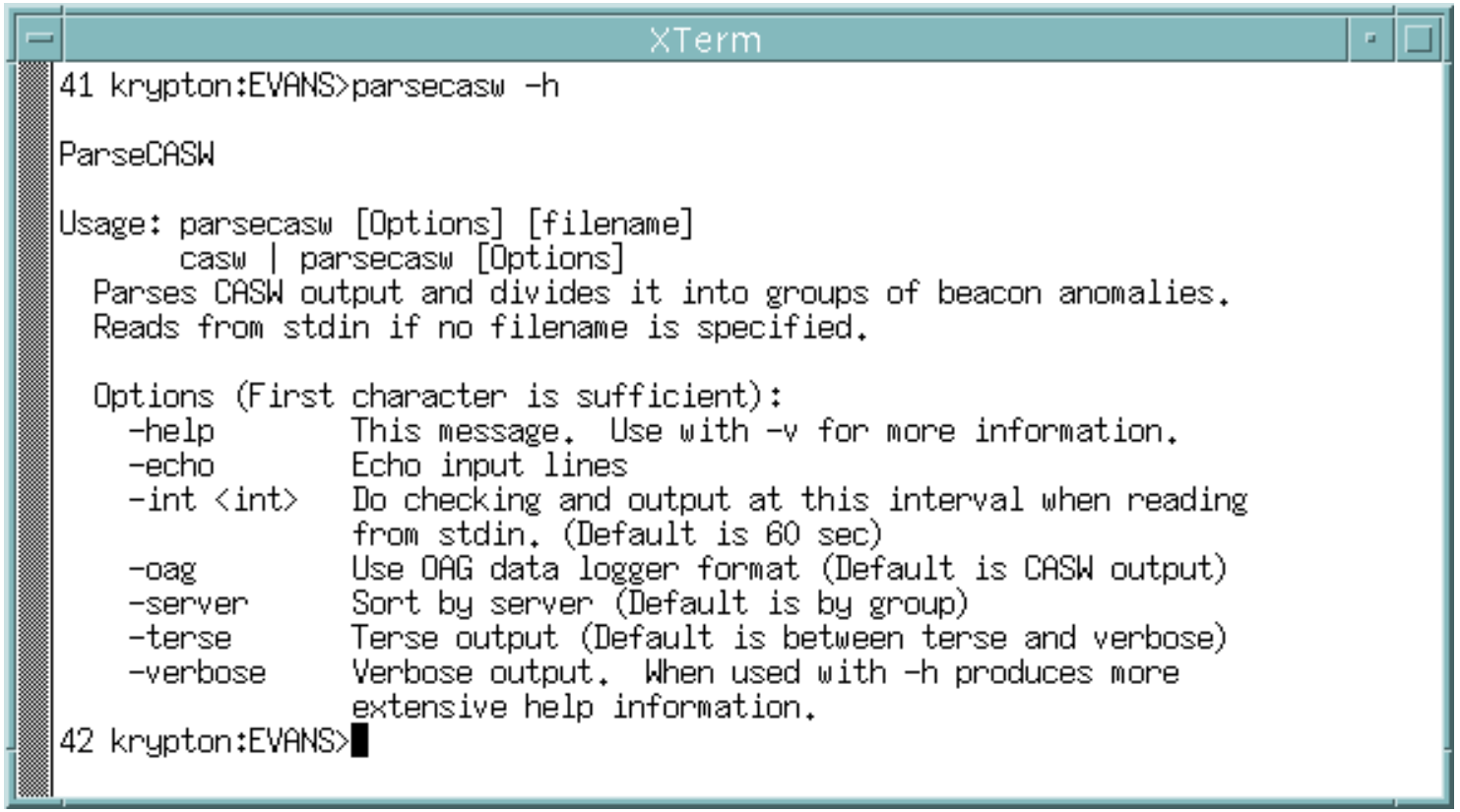


# ParseCASW Verbose Output

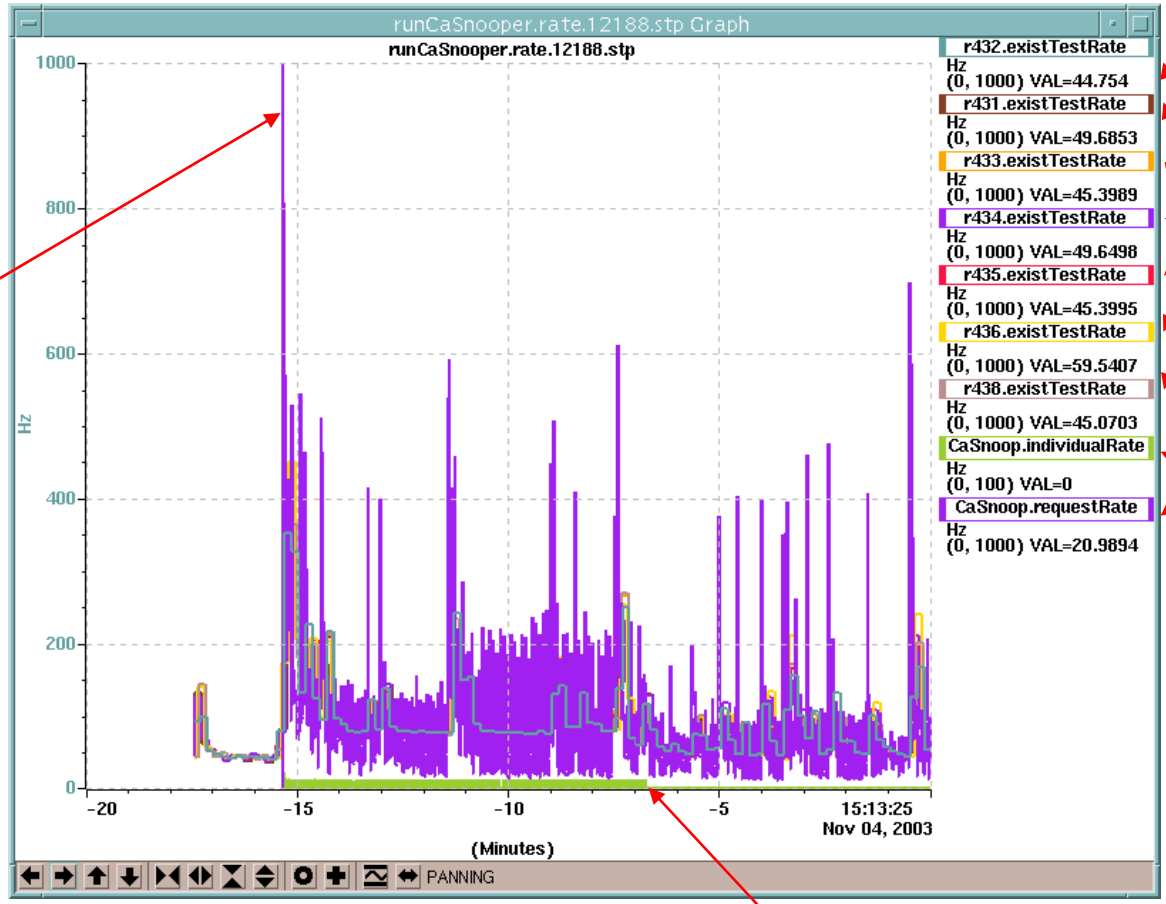
- **With the –v option you get more details**
  - You can decide for yourself what kind of event it is



# ParseCASW Options



# StripTool




CaSnooper was started here

Reverse Gateways (1 min update)

CaSnooper (1 sec update)

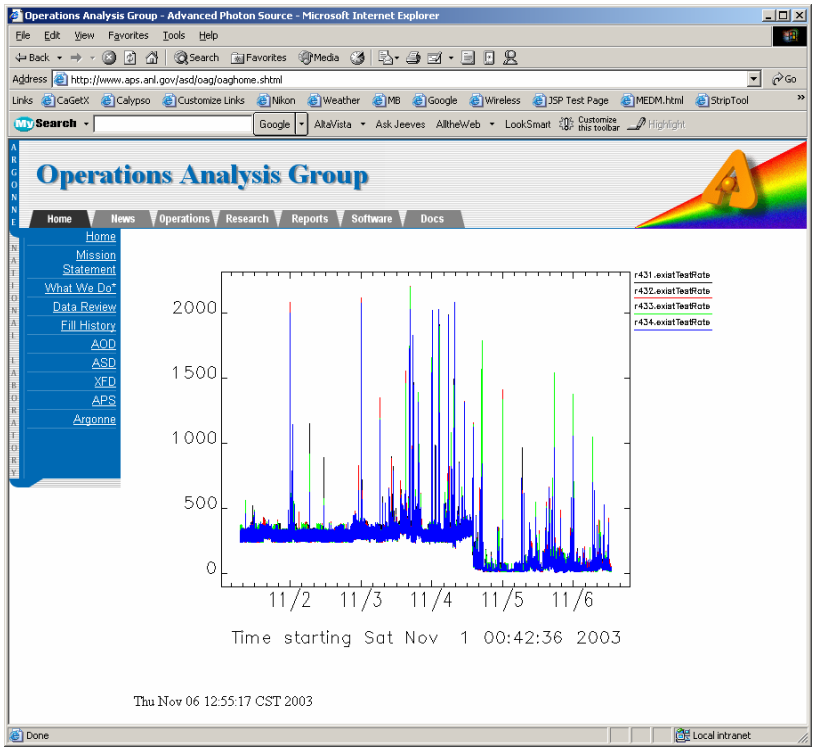
(Reverse Gateways, CaSnooper, and all IOCs see the same Search Requests)

Search for individual PV ended after about 8 minutes

# OAG Monitoring

- **The search request rates from the Reverse Gateways on the machine subnet are being continuously monitored**
- **You can can access the history from**

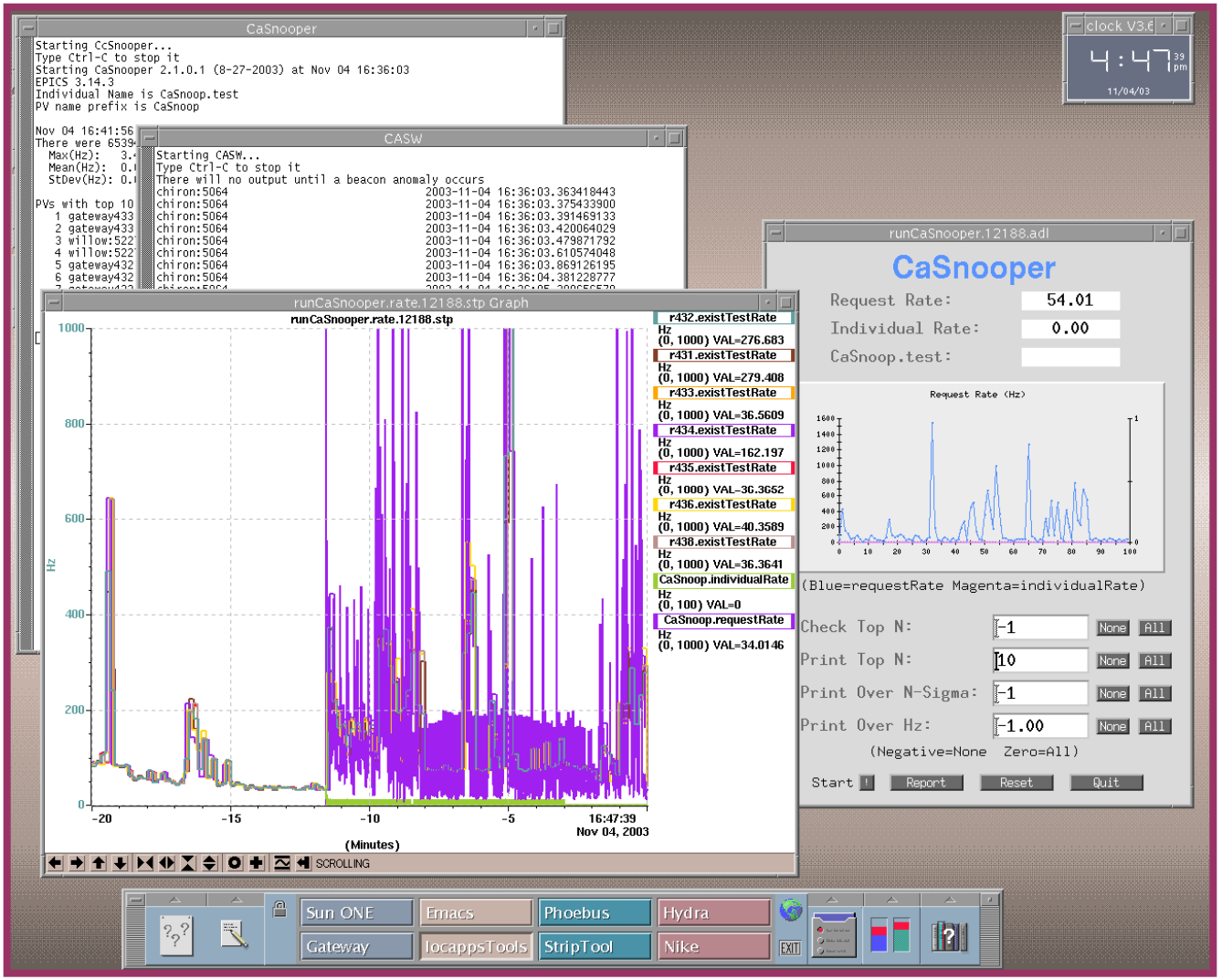
  http://www.aps.anl.gov/asd/oag/logging/MonitorDataReview.html



# *RunCaSnooper*

- **RunCaSnooper is a shell script that provides an interface to CaSnooper and CASW with an associated MEDM and StripTool**
- **Is APS specific, but the script can be modified for your situation**
  - Is part of IocappsTools (Type iocHelp for a list)
- **By default it brings up an MEDM and a StripTool and does not start CaSnooper**
  - If CaSnooper is running, the MEDM screen will not be white
  - If it is white, you can start CaSnooper from the MEDM screen
  - Uses EPICS_CA_REPEATER_PORT=9876 by default, not 5065
  - You can also start CASW and StripTool from the MEDM screen
- **Everything is generated on the fly and stored in /tmp**
- **Look in /tmp for:**
  - Logs of the CaSnooper reports and CASW output
  - MEDM ADL file and StripTool configuration files
- **Start it (type runCaSnooper) with no options:**
  - Displays extensive directions
  - Then optionally allows you to start it (type y to continue)

# RunCaSnooper



# RunCaSnooper Options

XTerm

```
48 krypton:EVANS>runCaSnooper -h

RunCaSnooper: Provides an interface to CaSnooper and CASW with an
associated MEDM and StripTool

Usage:
  runCaSnooper [Options]
  Options:
    -h           Help
    -d           Use no defaults, only the switches you enter
    +d           Use default setup without printing help
    -m           Do not start MEDM
    +m           Start MEDM [Default]
    -s           Do not start StripTool
    +s           Start StripTool [Default]
    -c           Do not start CaSnooper [Default]
    +c           Start CaSnooper
    -crp <int>   Set EPICS_CA_REPEATER_PORT for CaSnooper [Default is 5064]
                   Use a number greater than 5000, for example 6666
    -w           Do not start CASW [Default]
    +w           Start CASW
    -p <string> Specify a prefix for CaSnooper process variables
                   [Default is CaSnoop]
    -i <string> Specify an individual name for CaSnooper
                   [Default is CaSnoop.test]
    -clean       Remove caSnooper./* files created by runCaSnooper in /tmp
                   (This will also remove any log files created !)
```

# References


- **Jeffery O. Hill, *EPICS R3.14 Channel Access Reference Manual*, (EPICS Documentation, 2005 or latest).**

  http://www.aps.anl.gov/epics/modules/base/R3-14/index.php
- **W. Richard Stevens, *UNIX Network Programming*, (Prentice-Hall, Upper Saddle River, NJ, 1998) Vol. 1.**
- **K. Evans, Jr., *CaSnooper Reference Manual,* (EPICS Documentation, 2005 or latest).**

  http://www.aps.anl.gov/epics/extensions/caSnooper/index.php

# *Thank You*

*This has been an*

*APS Controls Presentation*

# *Thank You*

*This has been an*

*APS Controls Presentation*