OAK RIDGE National Laboratory

Channel Access

Kay Kasemir based on material by
Bob Dalesio (then LANL)
Ned Arnold (APS)
Ken Evans (APS)

July 2026

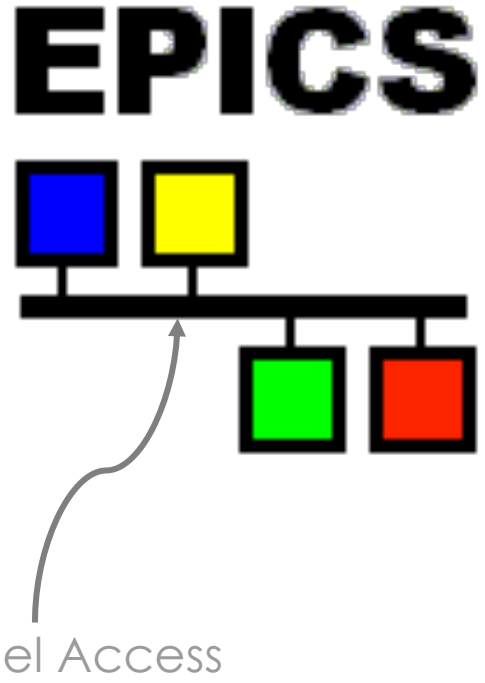
ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Channel Access: The EPICS Network Protocol

- Read and write Process Variables over the network.

- To many, CA is EPICS.
  - Especially if your background is w/ systems that have no IOC database.
  - "Integrate into EPICS" really means: Talk CA on the network.



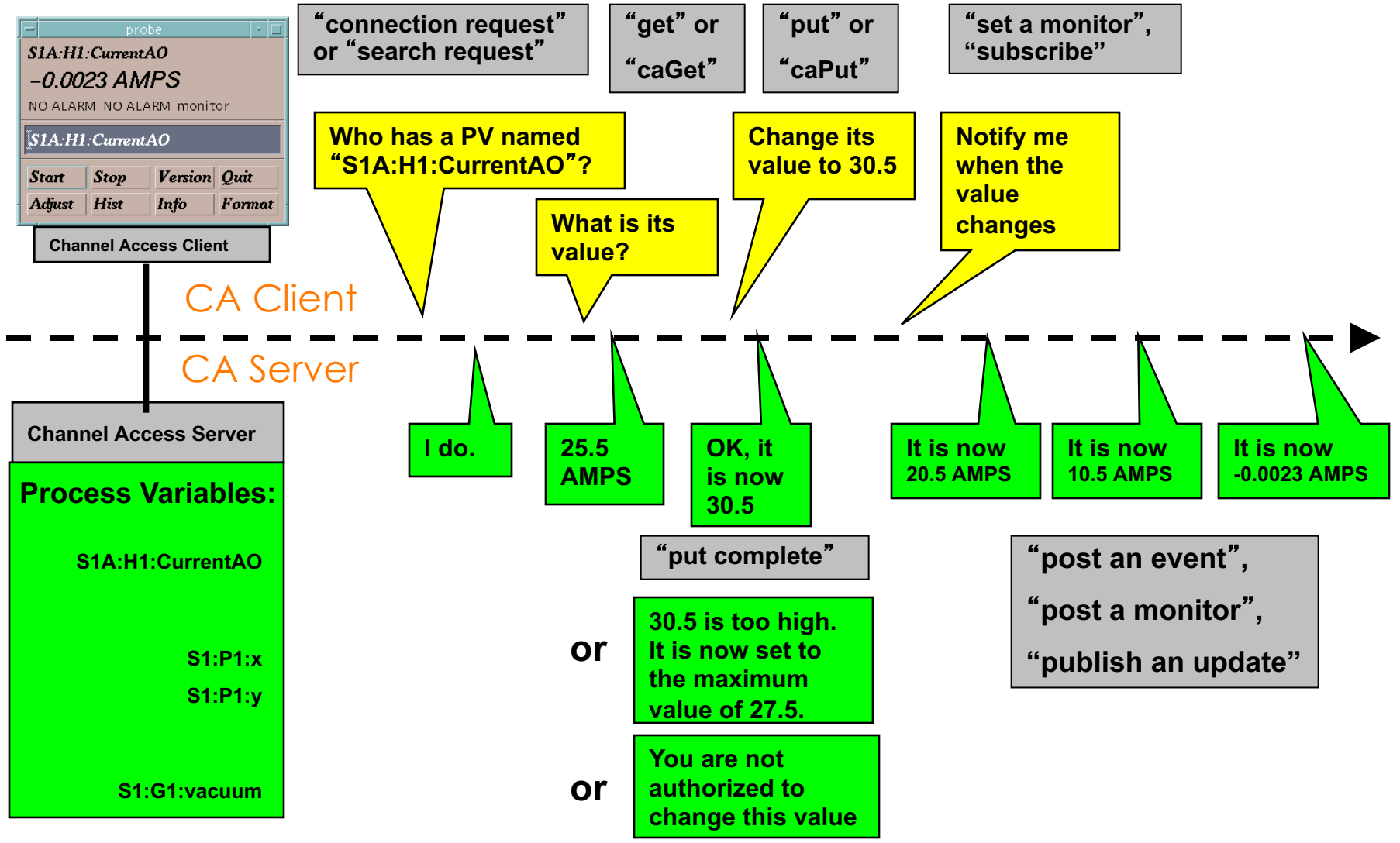
Advantages over similar Control System/IoT Protocols

- "Zero config" for small setups, yet scales to large networks
- Distributed, no central "Name Server" or "Broker" bottleneck
- Read/write/subscribe ('get', 'put', 'monitor')
  - Monitor sends initial value
  - 'put-callback' option for assured completion
- Data types that combine value with timestamp, units, status

To end user:

- Look at operator display and enjoy what you see
- Maybe use channel name to create your own displays, scripts, …

Channel Access in One Slide



4

Reading and Writing

'caget' command-line tool to read:

```
> caget epics-dev:tank
epics-dev:tank                  30.5382
> caget epics-dev:tank.DESC
epics-dev:tank.DESC             Tank Temperature or INVALID
```

'caput' to write:

```
> caput epics-dev:tank.DESC     "Tank temperature"
Old : epics-dev:tank.DESC       Tank Temperature or INVALID
New : epics-dev:tank.DESC       Tank temperature
```

Examples use /ics/examples/fishtank

Data types

'cainfo' to determine data type:

```
> cainfo epics-dev:tank
… Native data type: DBF_DOUBLE …

> cainfo epics-dev:tank.DESC
… Native data type: DBF_STRING …

> cainfo epics-dev:tank.SCAN
… Native data type: DBF_ENUM …
```

DBF_... DataBase Field

6

Data types…

'caget –d …' to fetch specific data type:

```
> caget -h
Usage: caget [options] <PV name> …
-d <type>: Request specific dbr type …(DBR_ prefix may be omitted)
…
```

```
> caget -d DOUBLE      epics-dev:tank
```

Just the value

```
> caget -d STS_DOUBLE  epics-dev:tank
```

```
> caget -d TIME_DOUBLE epics-dev:tank
```

Control system essentials

```
> caget -d GR_DOUBLE   epics-dev:tank
```

```
> caget -d CTRL_DOUBLE epics-dev:tank
```

Everything .. except the time??

```
> caget -d STRING      epics-dev:tank.SCAN
```

Enum combines a label with a number

```
> caget -d SHORT       epics-dev:tank.SCAN
> caget -d CTRL_ENUM   epics-dev:tank.SCAN
```

DBR_… DataBase Request

7

'camonitor'

'camonitor' *monitors* value changes:

```
> $ camonitor epics-dev:tank
epics-dev:tank                 2026-05-14 11:59:23.407152 68.6682
epics-dev:tank                 2026-05-14 11:59:23.907023 69.2315
epics-dev:tank                 2026-05-14 11:59:24.407060 69.7892
epics-dev:tank                 2026-05-14 11:59:24.906853 70.3413 HIGH MINOR
epics-dev:tank                 2026-05-14 11:59:25.406717 70.8879 HIGH MINOR
… plus one more each second…
… press Ctrl-C to stop …

> camonitor epics-dev:tank.DESC
epics-dev:tank.DESC            2026-05-14 11:20:40.311549 Tank temp
… and then nothing …
```

A.k.a. *subscribe*.

Channel Access supports monitoring

DBR_DOUBLE,
DBR_TIME_DOUBLE,
DBR_CTRL_DOUBLE, …

but ‘camonitor’ always uses DBR_TIME_...

8

When will ‘camonitor’ receive new value?

When the CA server (IOC) sends a new value!

- Binary records: Every change
- Analog records: VAL change beyond MDEL (monitor delta) or ADEL (archive delta)

```
$ camonitor -h
…
  -m <msk>: Specify CA event mask to use.
            <msk> is any combination of
            'v' (value), 'a' (alarm),
            'l' (log/archive), 'p' (property).
            Default event mask is 'va'
```

Client subscription options

- DBE_VALUE
  - Value changed beyond MDEL
- DBE_LOG
  - Meant for archive systems, … ADEL
- DBE_ALARM
  - Alarm status/severity changed
- DBE_PROPERTY
  - DBR_CTRL_... units, limits, … changed

Well behaved, generic clients

1. Determine the native *type*
2. Get DBR_CTRL_*type* once or monitor with DBE_PROPERTY
3. Monitor DBR_TIME_*type* with DBE_ALARM and either .._VALUE or .._LOG

→ Complete information while minimizing network traffic,

How do Clients find Channels?

.. and what happens then?

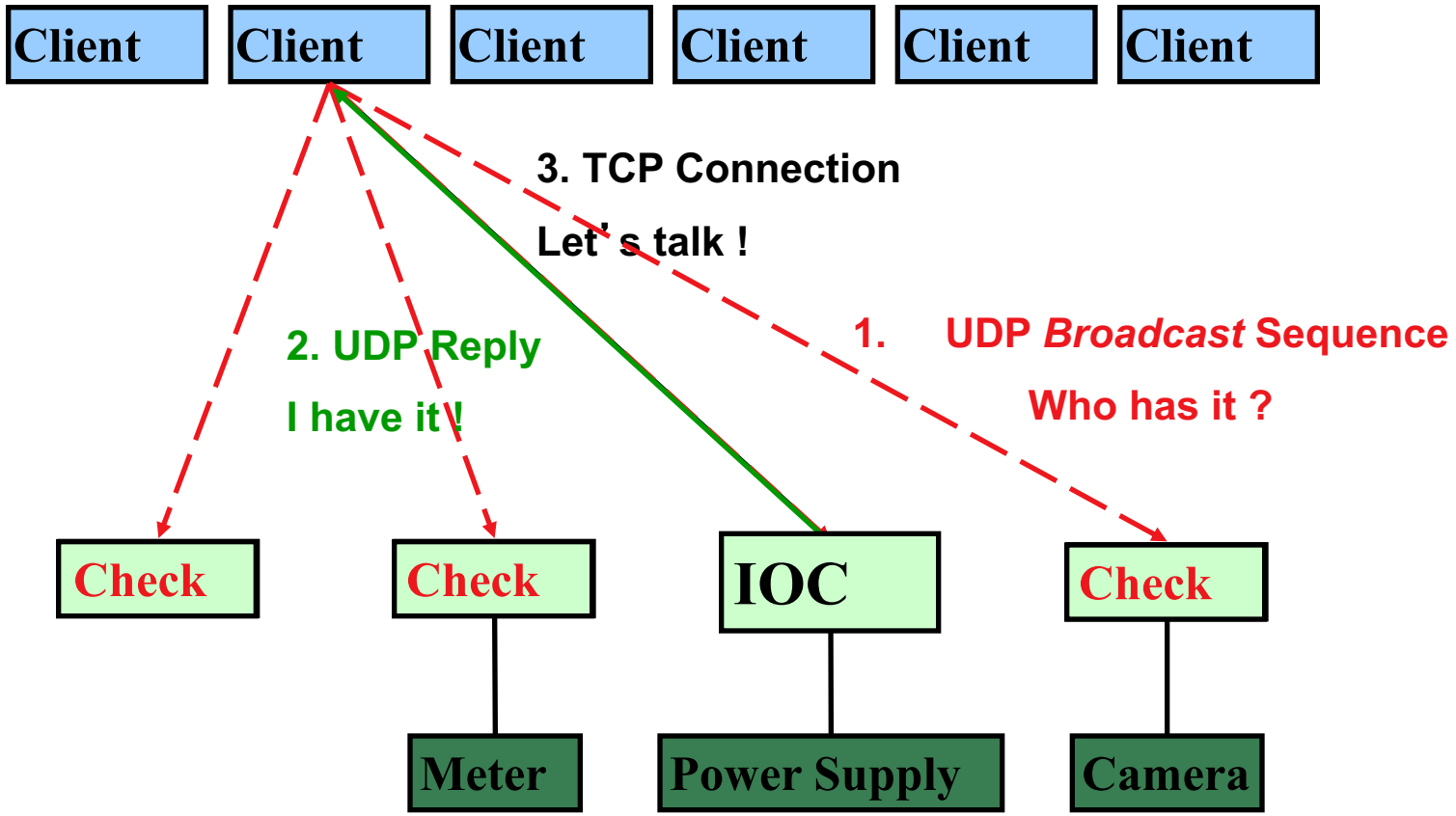
10

Internet 101

- User Datagram Protocol (UDP)
  - Sends a network packet
    - from one port on one computer
    - to one or more ports on one or more other computers.
    - ..with one or more listeners on the target ports
  - Fast!
  - Checksum: If the packet arrives, it's OK.
  - Not reliable: Packets get lost, arrive out-of-order, arrive more than once.
- Transmission Control Protocol (TCP)
  - Sends a stream of bytes from one port on one computer to another port on another computer, with exactly one listener on the target port
  - Reliable: Bytes arrive at the receiver in the correct order.
    - Basically, adds sequence numbers to UDP packets, requesting repeats for missing packages.
  - Slower, and message boundaries get lost:
    - "Hello Fred!" might arrive as "Hel" <pause> "lo F" <pause> "red!"

11

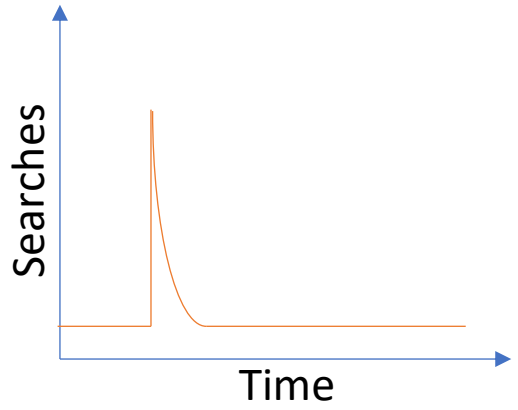
Search and Connect Procedure



12

Search Request

- A search request consists of a sequence of UDP packets
  - Per default: Broadcast to the local subnet.
    - Basically plug-and-play when you get started.
  - Or to IP addresses listed in EPICS_CA_ADDR_LIST
    - Most routers don't forward broadcasts!
    - You must add 'other' subnets or specific IOCs off the local subnet to that environment variable!
  - Starts with a small interval (0.1 s)
    - Doubles each time, until reaching 5 minute intervals.
    - Stops when it gets a response
    - Wakes again on "beacon anomaly" (details follow later)



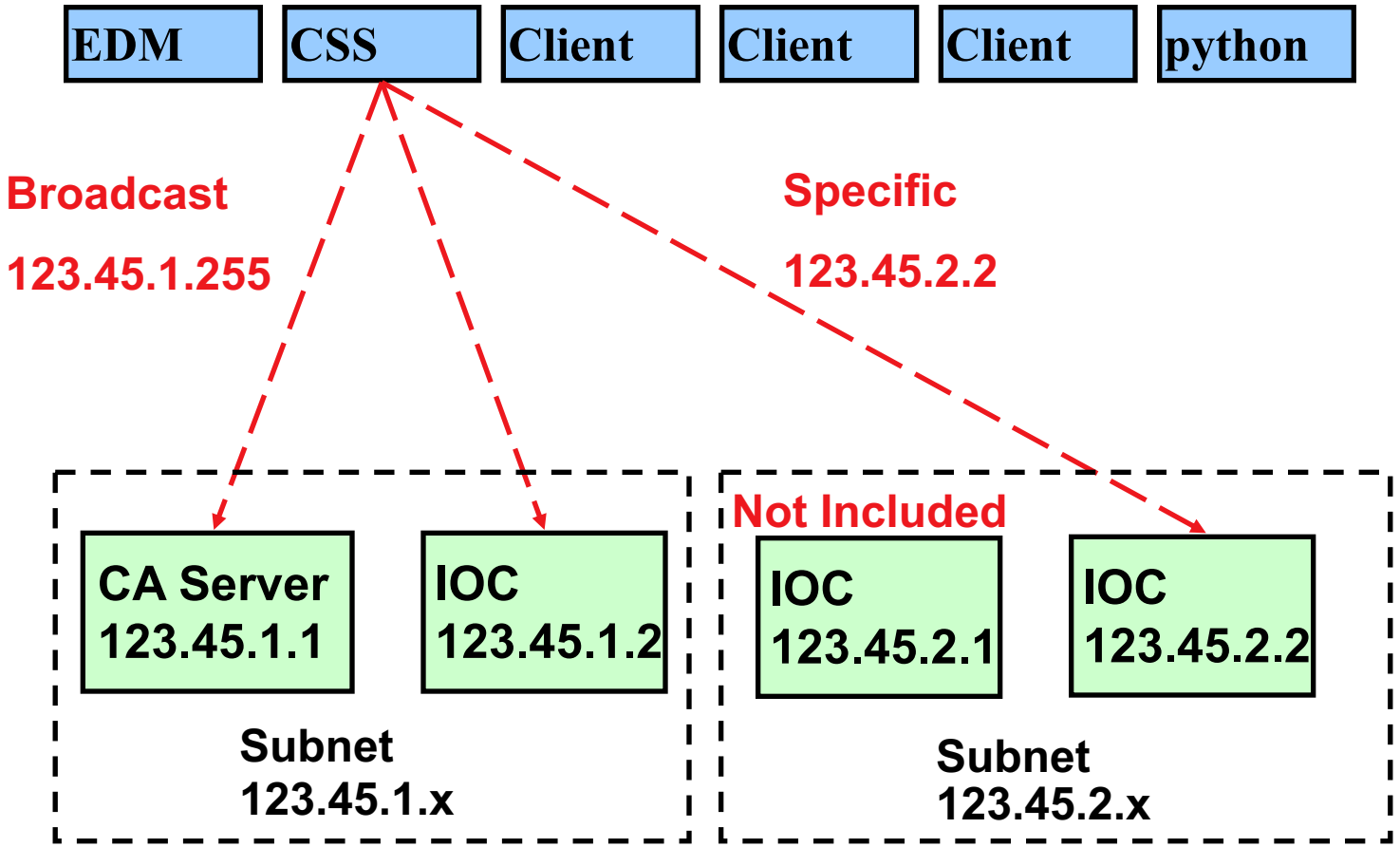
- CA Servers check each search packet
- Usually connects on the first packet or the first few
  - But non-existent PVs waste network bandwidth
  - Try to eliminate them!

13

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` or `ip address` to find it on Linux
- EPICS_CA_AUTO_ADDR_LIST
  - YES: Include default addresses in searches
  - NO: Do not search on default addresses, only use EPICA_CA_ADDR_LIST
  - If you set EPICS_CA_ADDR_LIST, usually set this to NO

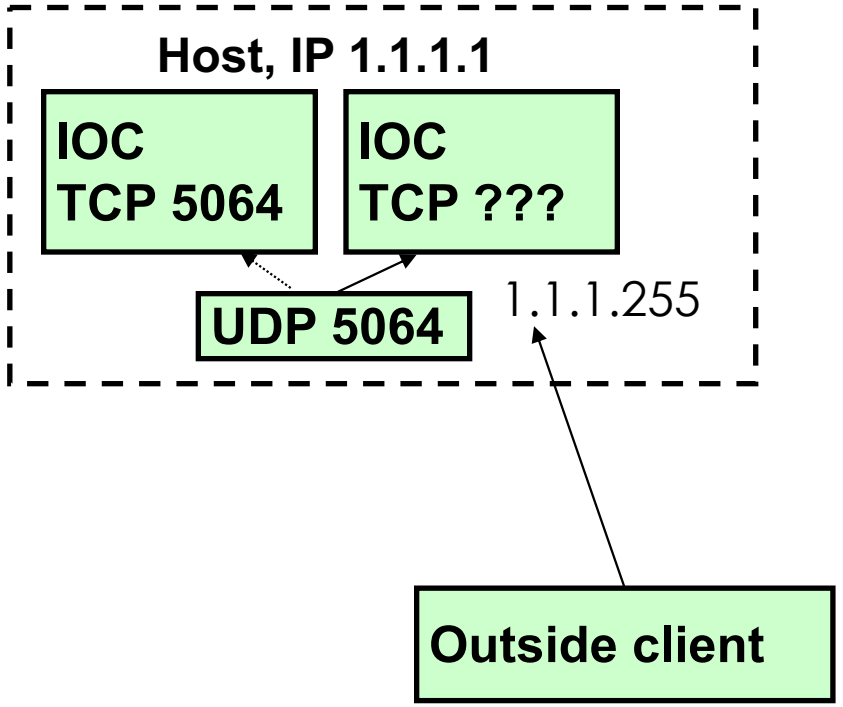
EPICS_CA_ADDR_LIST



15

Multiple "softIoc" on same Host

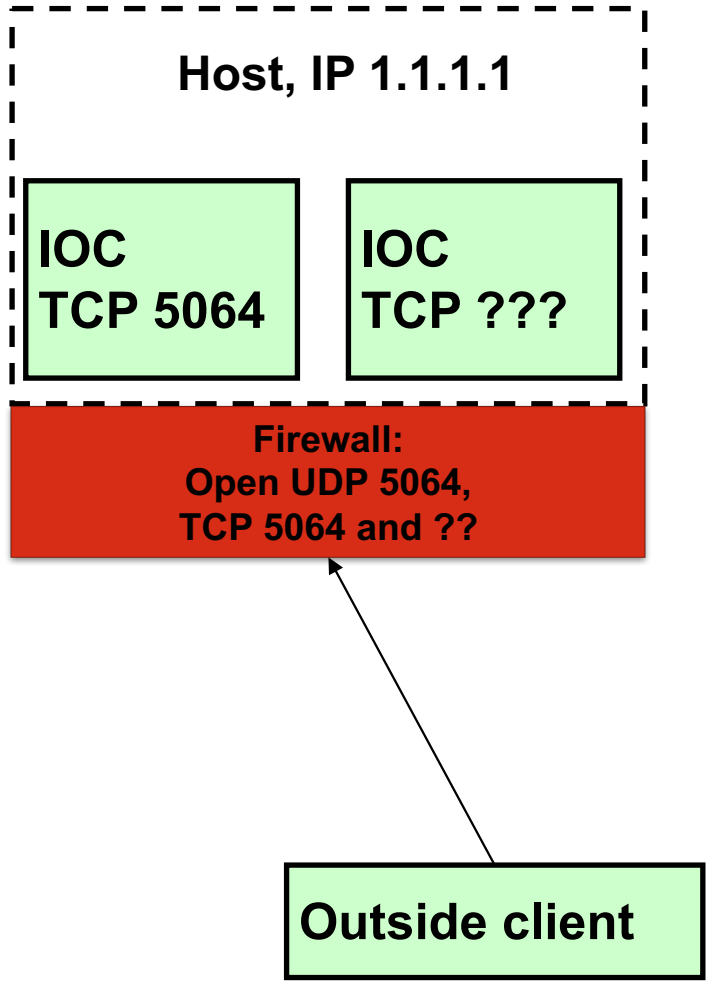
- IOCs on IP 1.1.1.1, net 1.1.1.0
  1. UDP 5064, TCP 5064
  2. UDP 5064, TCP ???
- Try to reach from other subnet
  - EPICS_CA_ADDR_LIST=1.1.1.1
    - Won't work!
    - Quirk in network kernels:
      Only the IOC started LAST (FIRST on Windows)
      will get anything on UDP 5064
  - EPICS_CA_ADDR_LIST=1.1.1.255
    - OK. When using broadcast into subnet, all IOCs on UDP 5064 will see search requests.



Firewall?!

- IOCs on IP 1.1.1.1, subnet 1.1.1.0
  - UDP 5064, TCP 5064
  - UDP 5064, TCP ???
  - EPICS_CA_ADDR_LIST=1.1.1.255
- Firewall cannot open unpredictable TCP ???
- Likely to block broadcasts

- Need to run CA Gateway:
  - Firewall allows access to CAGateway
  - CAGateway uses broadcast inside subnet



Handling of Network Interruptions

No Network is up 100%, so CA was designed to handle this:

- TCP connection closed by server?
  - Notify client code about problem
    - Operator displays tend to indicate this.
  - Client sends new search requests.
- No data nor beacon from server for 30 sec.?
  - Client sends "Are you there?" query
  - If no response for 5 sec, also notify client code, but TCP connection is kept open to avoid network storms.
  - If server eventually sends data: OK. Otherwise, we're waiting until the OS cuts the TCP connection (minutes to hours).

18

Beacons

- Assume all is fine, we are connected, but the data simply doesn't change.
  - How do we know the server is still OK?
- Assume we searched for a PV, didn't get any response for ~8 minutes.
  - How do we learn about a new CA server starting up which might have the missing PV? What triggers renewed search requests?

Beacons!

- UDP broadcast packet sent by a CA 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 (~2 ms)
  - Interval doubles each time until reaching 15 sec

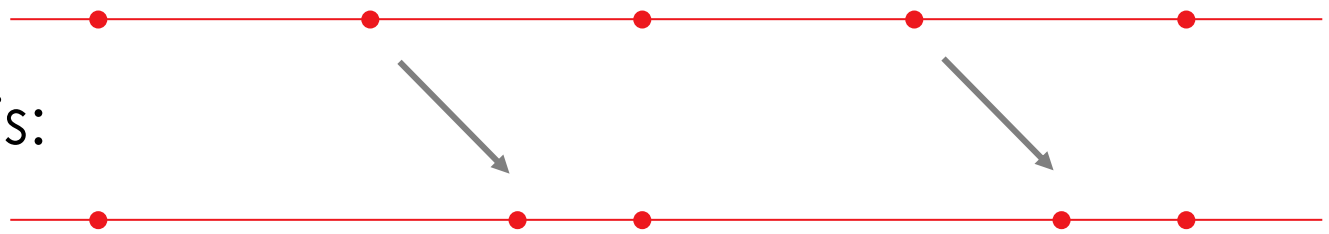
- Clients monitor the beacons
  - Receive beacons: Server is OK.
  - Receive new beacons at changing intervals: Beacon "anomaly", new CA server, restart searches.

Beacon: Seemed like a good idea, but…

- Archive setups may have many missing PVs

Actual Channels Count	90762
Actual Channels Connected	87240
Actual Channels Disconnected	3522

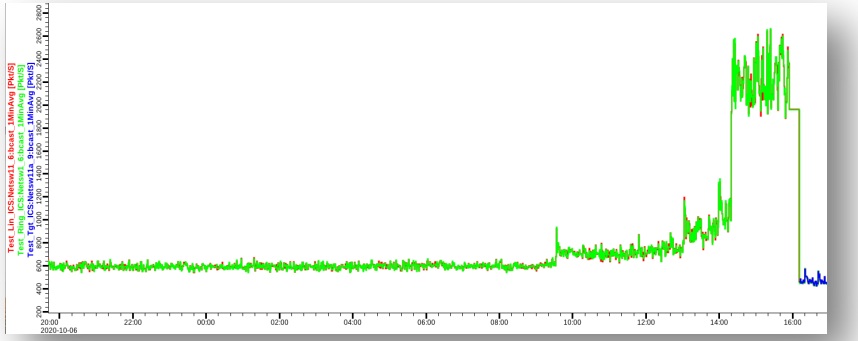
- Overloaded IOCs or network delays may change beacon pattern



.. into this:

➔ Clients with disconnected channels re-start their searches!

Burst in network traffic



Monitor & Maintain CA Network

- Check clients for disconnected channels
  - Archive
  - Displays
  - Run “caSnooper” to spot such clients

Actual Channels Count	90762
Actual Channels Connected	87240
Actual Channels Disconnected	3522

- Periodically run “casw”
  - Lists “beacon anomalies”
    Are those CA servers indeed “new”?

```
[ky9@ics-srv-accl2 src]$ casw
scl-vac-ioc1.ics.sns.gov:5064          2020-10-12 13:15:49.636783315
mebt-ps-ioc1.ics.sns.gov:5064          2020-10-12 13:16:35.517397056
dtl-rccs-ioc1.ics.sns.gov:5064         2020-10-12 13:16:35.525582171
mebt-ps-ioc1.ics.sns.gov:5064          2020-10-12 13:16:37.523308440
dtl-rccs-ioc1.ics.sns.gov:5064         2020-10-12 13:16:37.527365216
mebt-ps-ioc1.ics.sns.gov:5064          2020-10-12 13:16:39.517450221
dtl-vac-ioc2.ics.sns.gov:5064          2020-10-12 13:16:41.459136707
dtl-vac-ioc2.ics.sns.gov:5064          2020-10-12 13:16:46.522221982
dtl-vac-ioc2.ics.sns.gov:5064          2020-10-12 13:16:51.489182541
dtl-vac-ioc2.ics.sns.gov:5064          2020-10-12 13:16:56.355027915
mebt-ps-ioc1.ics.sns.gov:5064          2020-10-12 13:16:57.518379181
dtl-rccs-ioc1.ics.sns.gov:5064         2020-10-12 13:16:57.525735020
```

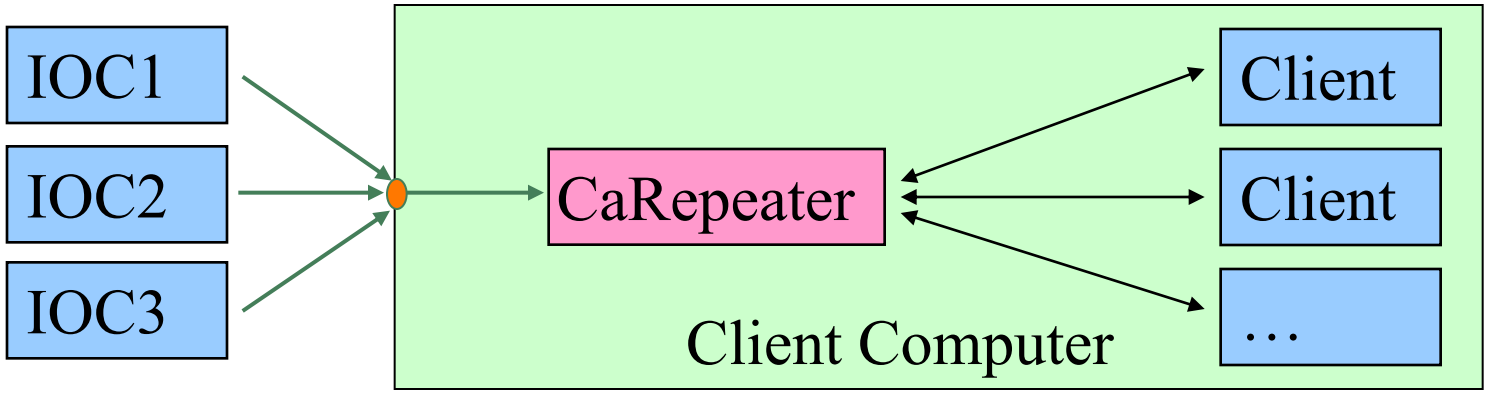
caSnooper Example

caSnooper –p20 –t20

```
Starting CaSnooper 2.1.2.3 (7-3-2013) at Oct 26 08:47:00
EPICS 3.14.8.2-SNS1
Individual Name is CaSnoop.test
Internal PV names are not being published

CaSnooper terminating after 20.00 seconds [0.33 minutes]
  Data collected for 20.00 seconds [0.33 minutes]

Oct 26 08:47:20:
There were 7967 requests to check for PV existence for 4825 different PVs.
  Max(Hz):   50.39
  Mean(Hz):  0.08
  StDev(Hz): 0.84

PVs with top 20 requests:
   1 ics-opi-ccr01.ics.sns.gov:48651 CF_ST:SFM_Rad_H:Alarm              50.39
   2 ics-opi-ccr01.ics.sns.gov:37071 DTL_RCCS:PT203:Pmp_In              23.20
   3 ics-opi-ccr01.ics.sns.gov:37071 DTL_RCCS:PT103:Pmp_In              15.60
   4 ics-srv-softioc3.ics.sns.gov:39989 @%b                              2.80
   5 ics-srv-softioc5.ics.sns.gov:55299 ICS_HPRF:IOC_linux_tunctrs:TimeShort 2.70
   6 ics-srv-arch1.ics.sns.gov:48219 GasInjectionLowFlow                1.95
   7 ics-srv-arch1.ics.sns.gov:33065 Cryo_Test:AlmModBeast              1.95
   8 ics-srv-arch1.ics.sns.gov:33065 Cryo_Beast:Health_Check            1.95
   9 ics-srv-arch1.ics.sns.gov:33065 CTF_ICS:PLC31:HBtOK                1.95
  10 ics-srv-arch1.ics.sns.gov:33065 CTF_ICS:PLC21:HBtOK                1.95
  11 ics-srv-arch1.ics.sns.gov:48219 storagetanklevellow                1.95
  12 mebt-ps-ioc1.ics.sns.gov:1039  IGNORE                             0.70
  13 ics-srv-softioc5.ics.sns.gov:48272 PPS_StackMonitor:IOC_Linux::Access_Cmd 0.60
  14 scl-ps-ioc9.ics.sns.gov:49262  IGNORE                             0.45
  15 ics-srv-cagate1-accl.ics.sns.gov:43293 COHERENT:LAr:PLC:TC2.DESC          0.30
  16 ics-srv-cagate1-accl.ics.sns.gov:43293 COHERENT:LAr:PLC:FC21.DESC         0.30
  17 ics-srv-cagate1-accl.ics.sns.gov:43293 COHERENT:LAr:GR:hum.DESC           0.30
  18 ics-srv-cagate1-accl.ics.sns.gov:43293 COHERENT:LAr:PLC:TE2.DESC          0.30
  19 ics-srv-cagate1-accl.ics.sns.gov:43293 COHERENT:LAr:GR:rtemp.DESC         0.30
  20 ring-diag-ioc-blm1.ics.sns.gov:1056 Ring_Diag:ND_B09:Fast1PulseLoss.VAL 0.30
```

caRepeater?

- Older OSs didn't allow multiple programs to listen to the same UDP port
  - They didn't see the beacons (UDP broadcasts)!
- 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]
  - .. and forwards them to clients.



24

Issues

- CA Client does not connect
  - Check basic network connectivity.
    - Can server and client machines 'ping' each other?
    - Check EPICS_CA_ADDR_LIST. Is server on different subnet?
- CA Client re-connects slowly after network issue or IOC reboot
  - Use casw, wireshark: Does the client computer receive the (anomal) beacons of the rebooting IOC?
  - Check EPICS_CAS_BEACON_ADDR_LIST,since routers will not forward beacons across subnets.
  - Check if 'caRepeater' is running on the client.

Points to remember

In 99% of the cases, CA "just works"

- If not, check EPICS_CA_ADDR_LIST
- If that's not it, there could be a subnet/router issue with UDP search broadcasts and beacons.

26

Comparison to PV Access

- Same basic idea: UDP searches, then TCP to get/put/monitor
- EPICS_PVA_ADDR_LIST instead of EPICS_CA_ADDR_LIST
- No caRepeater necessary
- More data types
  - "Normative types" for all the DBR_.. Info
  - Plus "Image", "Table", and optional custom types
- Supports IPv6
- "Secure PVA" development: Authentication, encryption