Channel Access Concepts

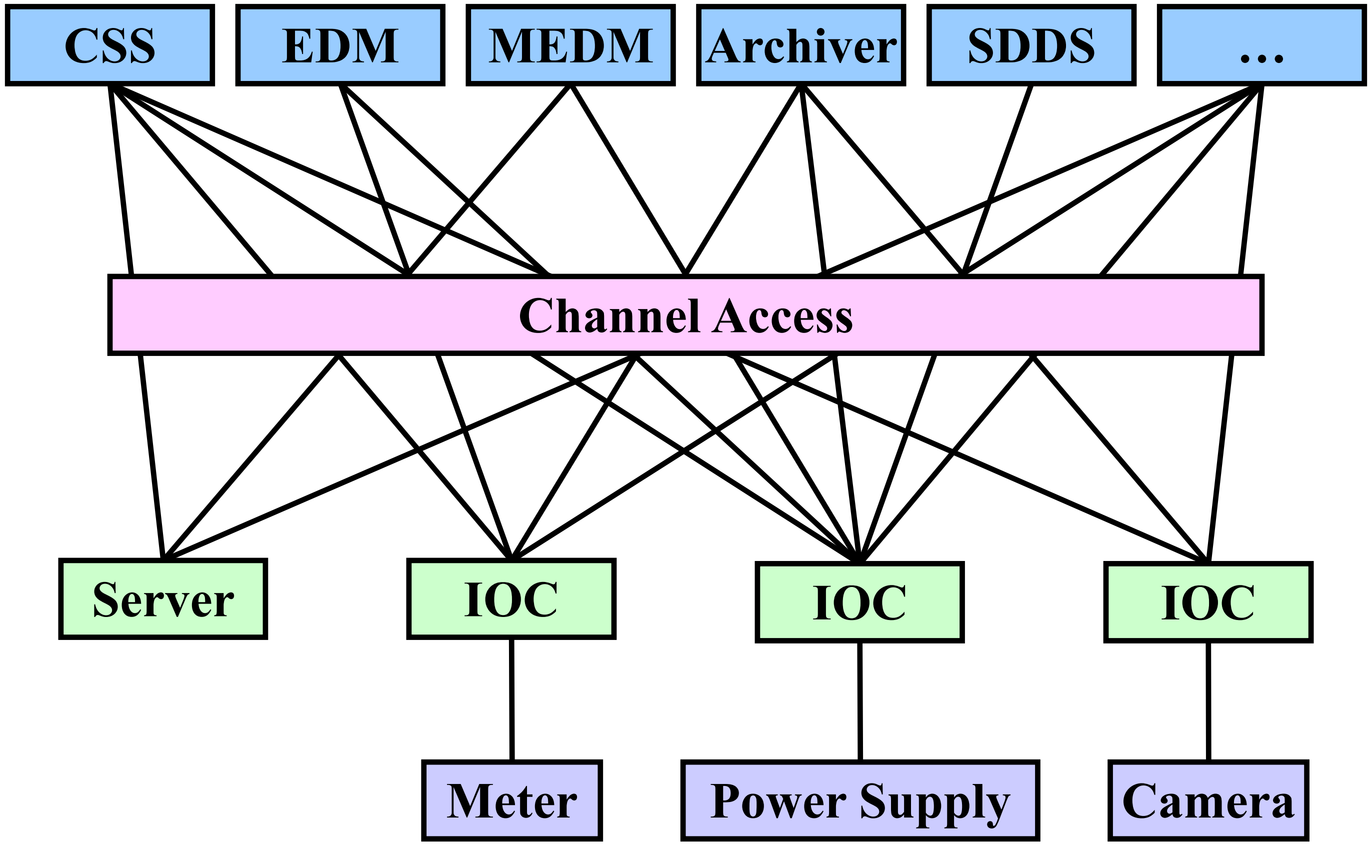
Andrew Johnson — AES-SSG, Argonne

Includes material from:

Ken Evans, Argonne

Kay Kasemir, ORNL

EPICS Overview



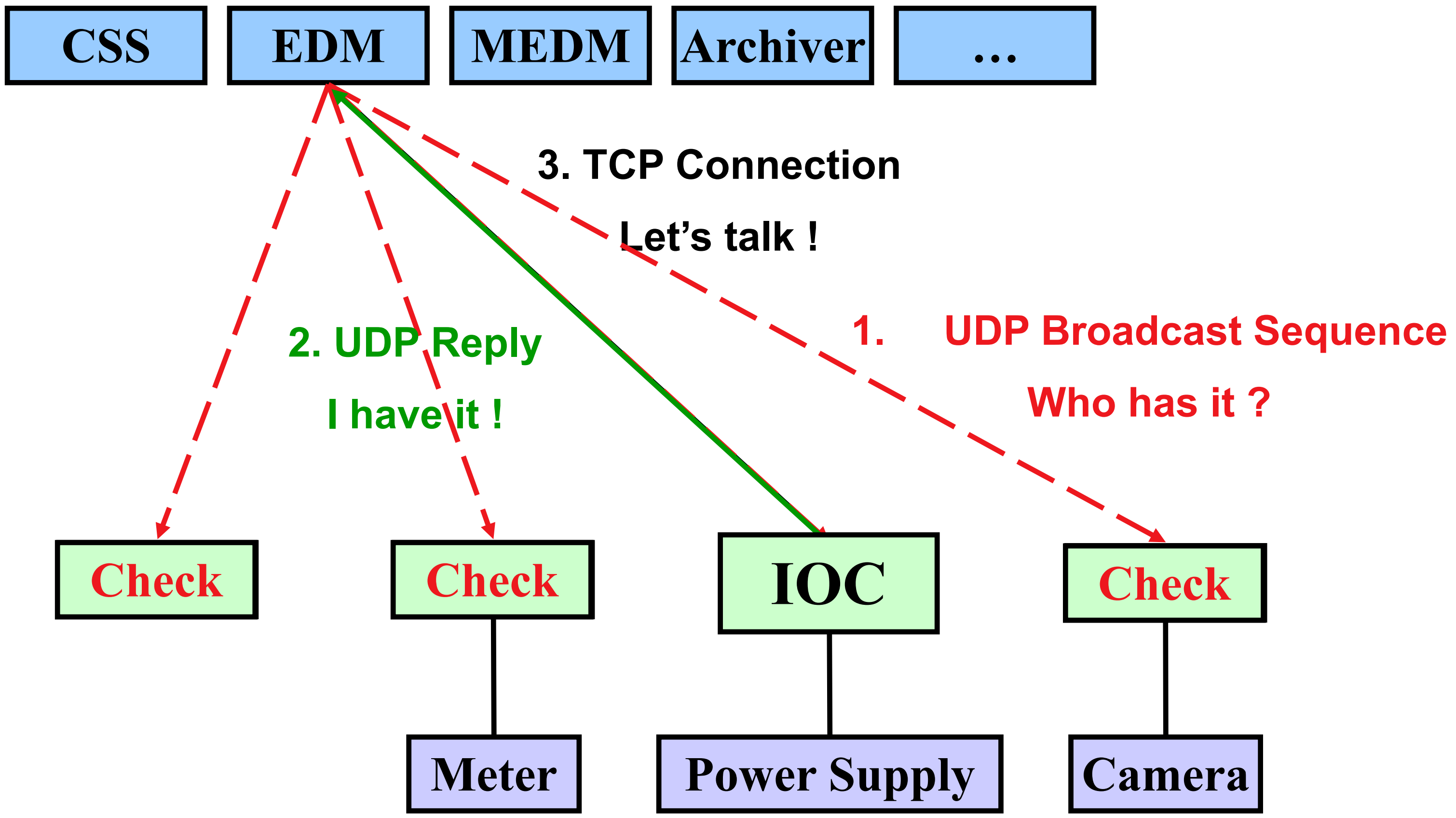
Channel Access Concepts

- Underlying Network Protocols
- Process Variable Connection Process
- Search Requests
- Beacons
- Beacon Anomalies
- Channel Disconnection
- CaRepeater
- Configuration Variables

Underlying Network Protocols

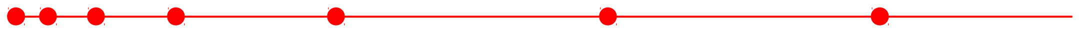
- Channel Access uses two Internet v4 protocols, UDP and TCP
- UDP (User Datagram Protocol)
  - Fast, one way message, unreliable
    - Packets may get lost, re-ordered, duplicated
  - Destination can be directed (unicast) or broadcast
    - Unicast: To a specific IP address, e.g. `123.45.6.100`
    - Broadcast: To all IP addresses in a subnet, e.g. `123.45.6.255`
  - Broadcasting across subnets is often restricted for security reasons
- TCP (Transmission Control Protocol)
  - Two way, reliable, persistent connection
    - Byte-streams are sent between the two end-points
  - OS handles acknowledgments, timeouts, retransmissions, etc.

Search and Connect Graphically



Search Requests

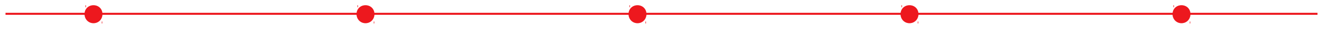
- A client makes a search request for each PV, to find its server
- Search requests for a PV start to be sent:
  - When a PV is first requested by a client
  - For unresolved PVs, whenever a beacon anomaly is seen or another PV is requested by the client application
- Search requests for multiple PVs are combined and sent over UDP
  - Initially repeated after 30 ms, the delay doubles each time until it reaches 5 seconds, where it stays



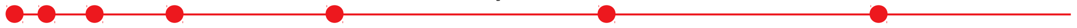
  - Searching stops as soon as a server responds
  - After 100 packets (about 8 minutes) searches are sent less frequently
  - The exact sequence may be different owing to fine tuning
- Clients usually connect on the first packet or within the first few
  - Requests for non-existent PVs can cause a lot of traffic

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 seconds by default



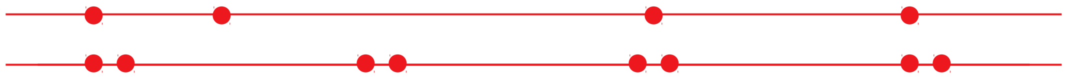
- When it starts up, each Server broadcasts a sequence of beacons
  - Starts with a small interval (25 ms, 75 ms for VxWorks)
  - Interval doubles each time
  - When it reaches 15 seconds, it stays there



  - Takes about 10 beacons or 40 seconds to get to steady state
- Clients monitor the beacons from all servers
  - Determine connection status, whether to reissue searches

Beacon Anomalies

- A Beacon Anomaly is any change from a normal beacon interval
  - "Normal" can be different for different servers
- If a client sees no beacons from a server it has channels from
  - After 30 sec the client sends a message over its TCP connection
  - If still no beacons and no reply from TCP, connection is down
  - Client program gets notified about each channel that disconnected
- Abnormal beacon interval:
  - Short: IOC has restarted
  - Long: IOC was disconnected
- Anomalies cause clients to retry any outstanding search requests
- Network problems can look like beacon anomalies



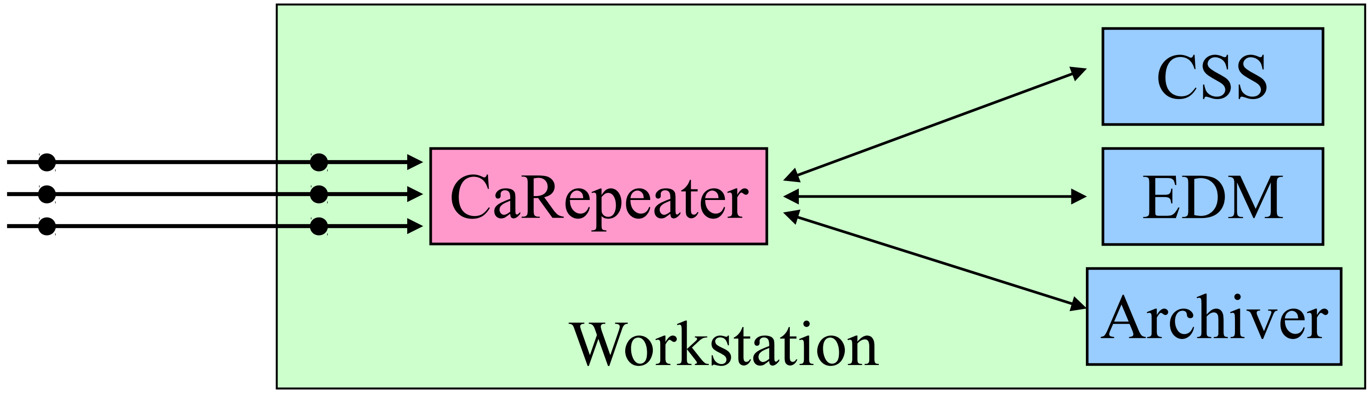
Channel Disconnection

- Behaviour of Base 3.14.5 and later
  - No response from server for 30 sec.
  - Client library sends an “Are you there” query
  - If no response within 5 sec, issue Virtual Circuit Disconnect
    - TCP socket connection is not closed by client library
    - TCP socket will eventually be closed by OS if no further traffic
  - Channels from this server marked disconnected (MEDM screens go white)
  - Library does not immediately reissue search requests
    - Helps recover from network disruptions
    - Searches will begin on next beacon anomaly

  - Clients that do not call `ca_poll()` frequently enough will get a virtual circuit disconnect, even though the server may still be OK
    - Clients written for 3.13 but using 3.14 sometimes have this problem
    - May be changed in future versions

CaRepeater

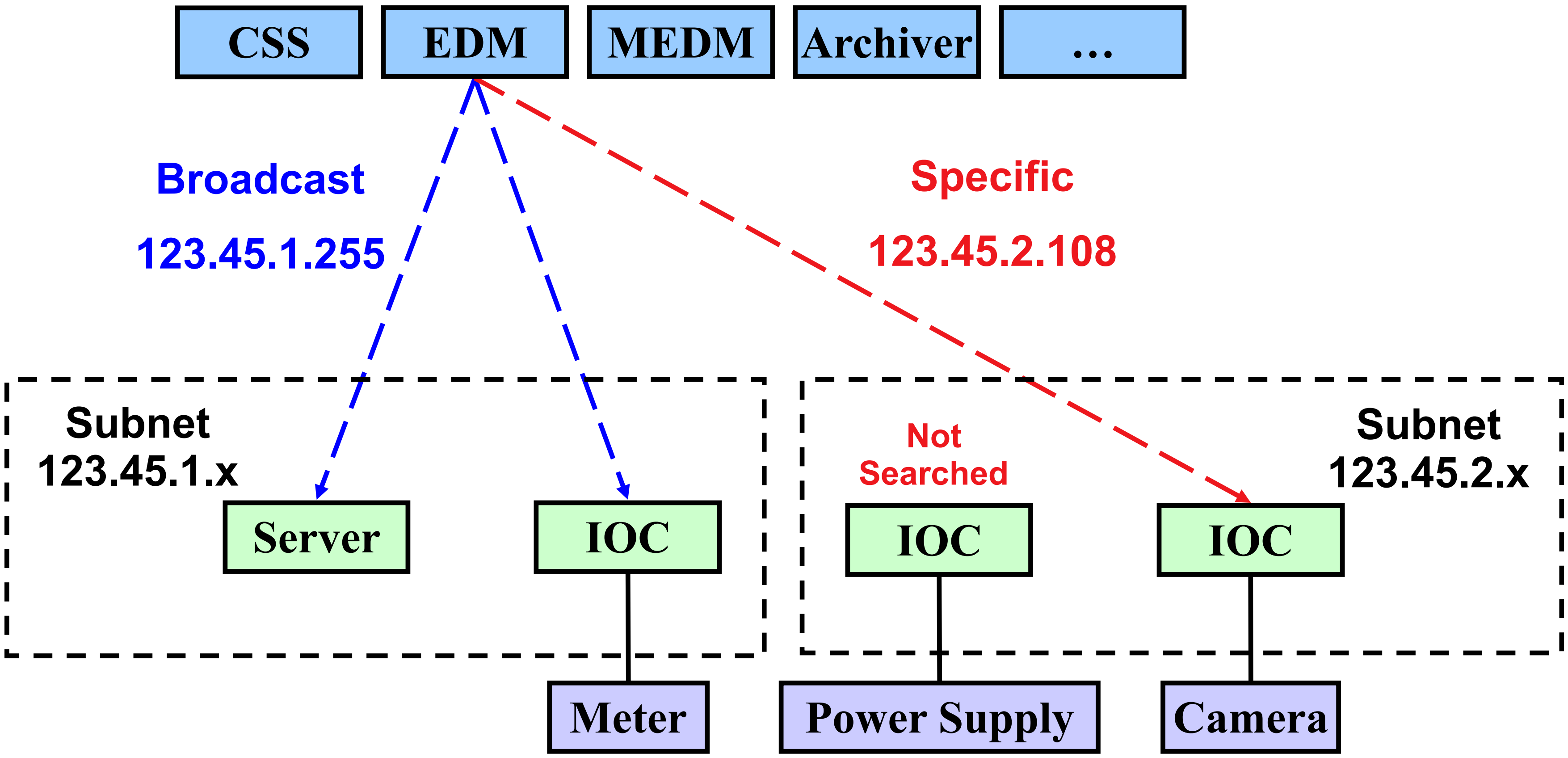
- When running multiple CA clients they all need to listen for beacons
  - UDP broadcasts are not normally copied to every process listening on the same UDP port
- The CaRepeater solves this problem
  - There is one CaRepeater process per workstation
  - Clients make a TCP connection to it when they start up
  - CaRepeater receives the beacons over UDP
    - **`EPICS_CA_REPEATER_PORT`** [usually 5065]
  - The CaRepeater forwards the beacons to its Clients over TCP



Important Environment Variables

- **`EPICS_CA_ADDR_LIST`**
  - Tells CA client library where to search for PVs
  - Is a list of IP addresses or hostnames (separated by spaces)
    - **`123.45.1.255 123.45.2.14 123.45.2.108`**
  - Default uses broadcast addresses of all interfaces on the workstation
    - Works fine when servers are all on same subnet as clients
  - Broadcast address
    - Search goes to all servers on the 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
  - Default is **`YES`**, if you set this at all it's usually 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
```

- CA Server (not IOC)

```
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

Reference Documentation

- Channel Access Reference Manual
  - Starting point for more information
  - Specific to each version of EPICS Base
    - Included with the Base source code
  - Also available from the EPICS website
    - EPICS Home → Base → R3.14 → R3.14.12

- A CA Protocol Description document exists
  - Created from the software in 2003 (R3.14.4), updated in 2008
  - Written by CosyLab, not by Jeff Hill
  - Covers most of the protocol semantics
  - Used to create the CAJ and JCAS Native Java library implementations
  - Updated version in development for Base R3.15

Summary

- Clients send search requests when they want a PV
- Each server checks if it has the PV for every packet in the search-request sequence
- Servers send beacons at regular intervals, with a faster pattern when they come up
- A beacon anomaly is any pattern that is not a regular beacon
- Beacon anomalies cause clients to resend search requests for any unresolved PVs
- Search requests end when a PV is found, but continue 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 use UDP port 5064, beacons use UDP port 5065