



# Wireshark CA Plug-in EPICS Channel Access Dissector

**Kazuro Furukawa, KEK**  
**Ron Rechenmacher, Fermilab**  
**Anze Zagar, Cosylab**  
**Klemen Zagar, Cosylab**

**Presented by**  
**Masanori Satoh, KEK**



# Background

## ◆ Ideas and efforts from several groups in the past

- ❖ Tech-talk proposal of CA Sniffer from Ned Arnold, APS
- ❖ Implementation of primary CA Plugin for Ethereal by Ron Rechenmacher, Fermilab
- ❖ (Managers love to have analyzers)

## ◆ KEK needed CA analyzer for efficient EPICS operation

- ❖ Without knowing above efforts
- ❖ Thought about Tcpdump extension for textual processing
- ❖ Discussion at ICALEPCS with Bob Dalesio and Jeff Hill
- ❖ Discussion with Ron Rechenmacher, Fermilab
- ❖ Implementation by Klemen and Anze Zagar, Cosylab



# CA Plug-in for Wireshark

## ◆ Wireshark (formally Ethereal)

❖ Is the most famous network protocol analyzer and is open source

✧ <http://www.wireshark.org/>

## ◆ Wireshark Plugin architecture

❖ EPICS channel access protocol dissection in CA plugin

✧ Development is well separated from main program

✧ Plugin distribution is simpler

◆ Only one file (shared/dinamic library file) for binary distribution

◆ One plugin directory and a simple patch (Makefile, etc) in a tar file for source





# CA Plug-in for Wireshark

## ◆ Graphical or Textual user interface

### ❖ Graphical interface for Online capture and Offline analysis

✧ With flexible filters

### ❖ Textual interface (tshark) for batch operation

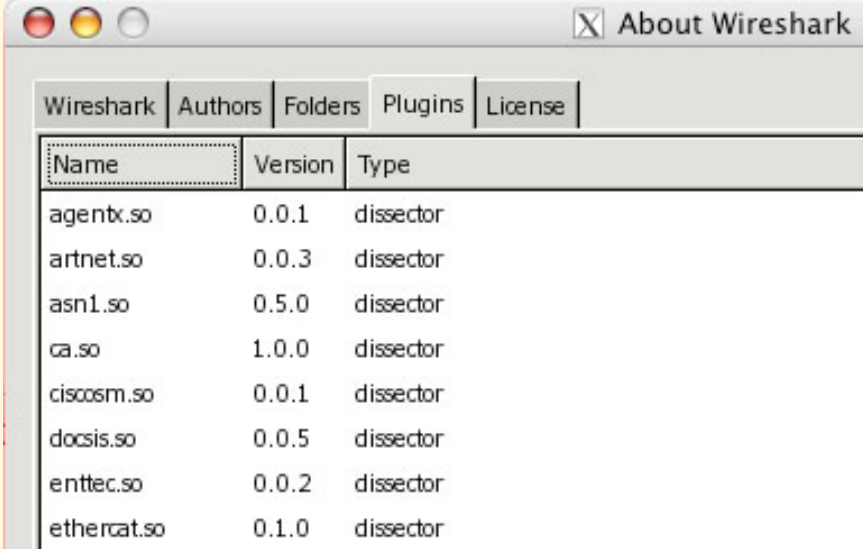
✧ Original intention at KEK was long-term rare event capturing and analysis

◆ Background operation was preferable

◆ Almost the same as tcpdump

◆ Captured data can be analyzed later

» With Graphical user interface



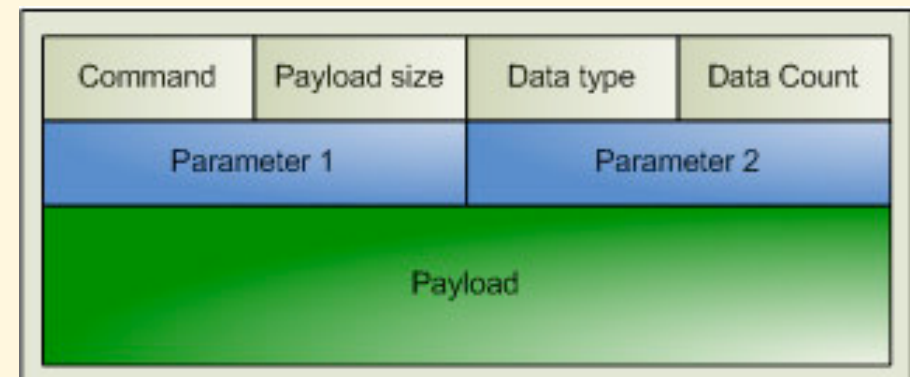
The screenshot shows the 'About Wireshark' dialog box with a tabbed interface. The 'Plugins' tab is selected, displaying a table of installed plugins.

Name	Version	Type
agentx.so	0.0.1	dissector
artnet.so	0.0.3	dissector
asn1.so	0.5.0	dissector
ca.so	1.0.0	dissector
ciscosm.so	0.0.1	dissector
docsis.so	0.0.5	dissector
enttec.so	0.0.2	dissector
ethercat.so	0.1.0	dissector



# CA Plugin

- ◆ **Dissects all CA packet header**
  - ❖ **Commands/replies and parameters**
    - ✧ **In Channel Access Protocol specification**
      - ◆ <http://epics.cosylab.com/cosyjava/JCA-Common/Documentation/CAproto.html>
- ◆ **Also tracks PV/Channel names along virtual circuit**
  - ❖ **Each packet only contains ID (CID/SID/SubscriptionID)**
    - ✧ **Indispensable for human-readable analysis**
- ◆ **Does not dissect payload**
  - ❖ **Use other EPICS tools**
    - ✧ **For data contents**





# Installation

## ◆ Binary installation

- ❖ Install normal Wireshark 0.99.8 or 0.99.7
- ❖ Install CA plugin binary
  - ✧ From <http://www-linac.kek.jp/cont/epics/wireshark/>
  - ✧ Windows, Linux, MacOSX (x86/ppc) for now

## ◆ Building from source

- ❖ Get Wireshark (0.99.8 or 0.99.7)
- ❖ Expand CA plugin source
- ❖ Apply patch
- ❖ Normal building procedure
  - ✧ <http://www-linac.kek.jp/cont/epics/wireshark/> for details
  - ✧ Gtk+ and packet capture software are required



# Simple Usage for EPICS

## ◆ Invoke Wireshark

## ◆ Capture options

❖ Capture Filter: “port 5064 or port 5065”

## ◆ Start capture

## ◆ (Stop capture)

## ◆ Apply display/analysis filter

### ❖ Filter examples

✧ **ca.cmd==1**

◆ Symbolic names like CA\_PROTO\_SEARCH in Helper

✧ **ca.chanName=="fred" or ca.channel=="fred"**

◆ Packets related to a PV named fred

✧ **ca.channel matches “^VAC:IP.\*:Pressure”**

✧ **ca.channel contains “VAC:IP”**

◆ PV name string or regular-expression matching

```
ca.cmd - CA Command ID
ca.paySz - CA Payload size
ca.type - CA Data type
ca.cnt - CA Data Count
ca.p1 - CA Parameter 1
ca.p2 - CA Parameter 2
ca.tcpPort - TCP port of responding server
ca.srvrId - Temporary SID
ca.chnId - Channel CID
ca.minorVer - Minor protocol version
ca.srvrVer - Server protocol version
ca.desiredPrio - Desired Priority
ca.userName - User name
ca.hostName - Host name
ca.chanName - Channel name
ca.accRights - Access Rights
ca.ioid - Client provided IOID
ca.subscrptId - Client provided Subscription ID
ca.evLo - Low value
ca.evHi - High value
ca.evTo - To value
ca.evMonMsk - Monitor mask
ca.status - Status
ca.reply - Reply
ca.reserved - Reserved (Should be zero)
ca.unused - Unused
ca.clientip - Client IP address
ca.serverip - Server IP address
ca.repeaterip - Repeater IP address
ca.strDat - String data
ca.dbIDat - Double prec.float data
ca.deprecated - Obsolete (Obsolete)
ca.data - data (formatted data)
ca.zero - zero (should be zero)
ca.undecoded - undecoded (Yet undecoded by dissector)
ca.channel - Corresponding channel
```



# Selecting EVENT\_ADD command/response

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: `ca.cmd == 1` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
99	10.177276	cabbage	abco4.kekb.kek.jp	CA tcp	5064 > 49208 Responses: CA_PROTO_EVENT_ADD
109	10.800465	abco4.kekb.kek.jp	IOCBMCCC	CA tcp	33184 > 5064 Requests: CA_PROTO_EVENT_ADD, CA
111	10.800816	IOCBMCCC	abco4.kekb.kek.jp	CA tcp	5064 > 33184 Responses: CA_PROTO_EVENT_ADD, CA
113	11.023946	cabbage	abco4.kekb.kek.jp	CA tcp	5064 > 49208 Responses: CA_PROTO_EVENT_ADD
115	11.055417	cabbage	abco4.kekb.kek.jp	CA tcp	5064 > 49208 Responses: CA_PROTO_EVENT_ADD, CA
117	11.089468	cabbage	abco4.kekb.kek.jp	CA tcp	5064 > 49208 Responses: CA_PROTO_EVENT_ADD, CA
119	11.158231	cabbage	abco4.kekb.kek.jp	CA tcp	5064 > 49208 Responses: CA_PROTO_EVENT_ADD
121	11.175858	cabbage	abco4.kekb.kek.jp	CA tcp	5064 > 49208 Responses: CA_PROTO_EVENT_ADD
123	11.742064	IOCBMCCC	abco4.kekb.kek.jp	CA tcp	5064 > 33184 Responses: CA_PROTO_EVENT_ADD, CA
125	12.025823	cabbage	abco4.kekb.kek.jp	CA tcp	5064 > 49208 Responses: CA_PROTO_EVENT_ADD

CA Command ID: CA\_PROTO\_EVENT\_ADD (1)  
CA Payload size: 24  
CA Data type: DBR\_TIME\_DOUBLE (20)  
CA Data Count: 1  
Status: ECA\_NORMAL (1)  
Client provided Subscription ID: 0x00000002  
data  
Corresponding channel: CO\_IOC:EMCCC:HB

EPICS Channel Access, tcp Response, cmd: 1 (CA\_PROTO\_EVENT\_ADD) 172.19.51.121:5064 -> 172.19.57.152:33184

```
0000 00 13 72 40 8a 00 00 01 af 14 55 0f 08 00 45 00  ..r@... ..U...E.
0010 00 a4 47 6a 40 00 40 06 2d b2 ac 13 33 79 ac 13  ..Gj@.@. -...3y..
0020 39 98 13 c8 81 a0 0c 0d 30 25 61 1f bc 75 80 18  9..... 0%.a.u..
0030 20 00 80 57 00 00 01 01 08 0a 00 16 f2 f8 2e 74  ..W.... ..t
0040 f8 12 00 01 00 18 00 14 00 01 00 00 00 01 00 00  .....
```

Corresponding channel (ca. Packets: 236 Displayed: 112 Marked: 0 Dropped: 0





# Selecting “fred” related packets

The screenshot shows the Wireshark interface with the following details:

- Window title: eth1: Capturing - Wireshark
- Filter: `ca.channel=="fred" or ca.chanName=="fred"`
- Packet list table:

No.	Time	Source	Destination	Protocol	Info
367	17.787231	172.19.64.37	172.19.95.255	CA udp	33240 > 5064 Requests: CA_PROTO_VERSION, CA_PROTO_SEARCH,
372	17.788544	172.19.64.37	172.19.64.44	CA tcp	
1122	46.427100	172.19.64.37	172.19.95.255	CA udp	39131 > 5064 Requests: CA_PROTO_VERSION, CA_PROTO_SEARCH,
1127	46.430746	172.19.64.37	172.19.64.44	CA tcp	40475 > 5064 Requests: CA_PROTO_VERSION, CA_PROTO_CLIENT
1129	46.430976	172.19.64.44	172.19.64.37	CA tcp	5064 > 40475 Responses: CA_PROTO_VERSION
1130	46.431208	172.19.64.44	172.19.64.37	CA tcp	5064 > 40475 Responses: CA_PROTO_ACCESS_RIGHTS, CA_PROTO_C
1133	46.431742	172.19.64.37	172.19.64.44	CA tcp	40475 > 5064 Requests: CA_PROTO_EVENT_ADD, CA_PROTO_EVENT
1134	46.433220	172.19.64.44	172.19.64.37	CA tcp	5064 > 40475 Responses: CA_PROTO_EVENT_ADD, CA_PROTO_EVENT
1137	46.529256	172.19.64.44	172.19.64.37	CA tcp	5064 > 40475 Responses: CA_PROTO_EVENT_ADD
1141	46.629421	172.19.64.44	172.19.64.37	CA tcp	5064 > 40475 Responses: CA_PROTO_EVENT_ADD
1144	46.729226	172.19.64.44	172.19.64.37	CA tcp	5064 > 40475 Responses: CA_PROTO_EVENT_ADD
1147	46.829318	172.19.64.44	172.19.64.37	CA tcp	5064 > 40475 Responses: CA_PROTO_EVENT_ADD
1153	46.929475	172.19.64.44	172.19.64.37	CA tcp	5064 > 40475 Responses: CA_PROTO_EVENT_ADD

Packet details for Frame 1137:

- Frame 1137 (106 bytes on wire, 106 bytes captured)
- Ethernet II, Src: Dell\_cd:07:aa (00:11:43:cd:07:aa), Dst: Intel\_d5:3a:bc (00:02:b3:d5:3a:bc)
- Internet Protocol, Src: 172.19.64.44 (172.19.64.44), Dst: 172.19.64.37 (172.19.64.37)
- Transmission Control Protocol, Src Port: ca-1 (5064), Dst Port: 40475 (40475), Seq: 161, Ack: 185, Len: 40
- EPICS Channel Access, tcp Response, cmd: 1 (CA\_PROTO\_EVENT\_ADD) 172.19.64.44:5064 -> 172.19.64.37:40475
  - CA Command ID: CA\_PROTO\_EVENT\_ADD (1)
  - CA Payload size: 24
  - CA Data type: DBR\_TIME\_DOUBLE (20)
  - CA Data Count: 1
  - Status: ECA\_NORMAL (1)
  - Client provided Subscription ID: 0x00000001
  - data
  - Corresponding channel: fred

Hex dump:

```
0000 00 02 b3 d5 3a bc 00 11 43 cd 07 aa 08 00 45 00 ..... C....E.
0010 00 5c 21 f1 40 00 40 06 40 33 ac 13 40 2c ac 13 .\!.@.@.#3.@...
0020 40 25 13 c8 9e 1b 7c 59 26 83 e7 a5 83 c6 80 18 @%....|Y &.....
0030 16 a0 aa e3 00 00 01 01 08 0a 60 b7 04 18 66 88 ..... ..f.
0040 91 52 00 01 00 18 00 14 00 01 00 00 00 01 00 00 .R.....
0050 00 01 00 00 00 00 22 30 c7 09 08 73 5c 50 00 00 ..... "O ...s\P..
0060 00 00 40 01 11 8c 20 00 00 00 .....@....
```



## Hints

- ◆ **Combination with CA Snooper may enhance network trouble-shooting**
- ◆ **Expression button helps filter expression construction**
- ◆ **tshark may be used to capture packets, and later Wireshark can be used to analyze them**
- ◆ **Data contents dissection necessary?**



## Summary

- ◆ **Wireshark CA plugin was build with efforts by many people**
- ◆ **It may be used for the efficient operation of EPICS system and for the trouble-shooting**
- ◆ **Please send any comments to**
  - ❖ **< kazuro.furukawa @ kek.jp >**



**Thank you**