OAK RIDGE
National Laboratory

EPICS and IPv6

Kay Kasemir
Michael Davidsaver

April 2023 EPICS Meeting

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

U.S. DEPARTMENT OF
ENERGY

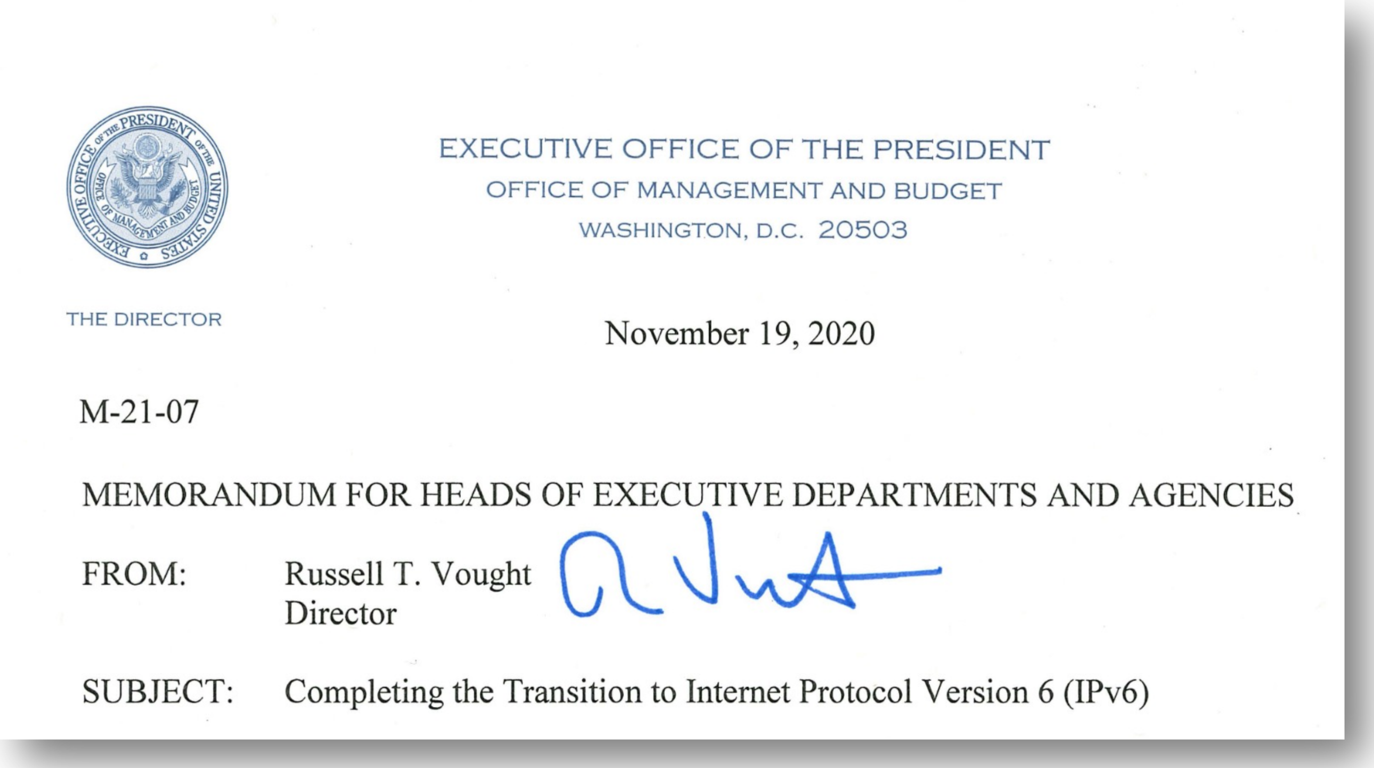
Why?

- *Most recent internet protocol*

  Introduced 1995,
  Internet Standard 2017

- *IPv4 addresses are limited*

- *IPv6 transition plans*



EXECUTIVE OFFICE OF THE PRESIDENT
OFFICE OF MANAGEMENT AND BUDGET
WASHINGTON, D.C. 20503

THE DIRECTOR

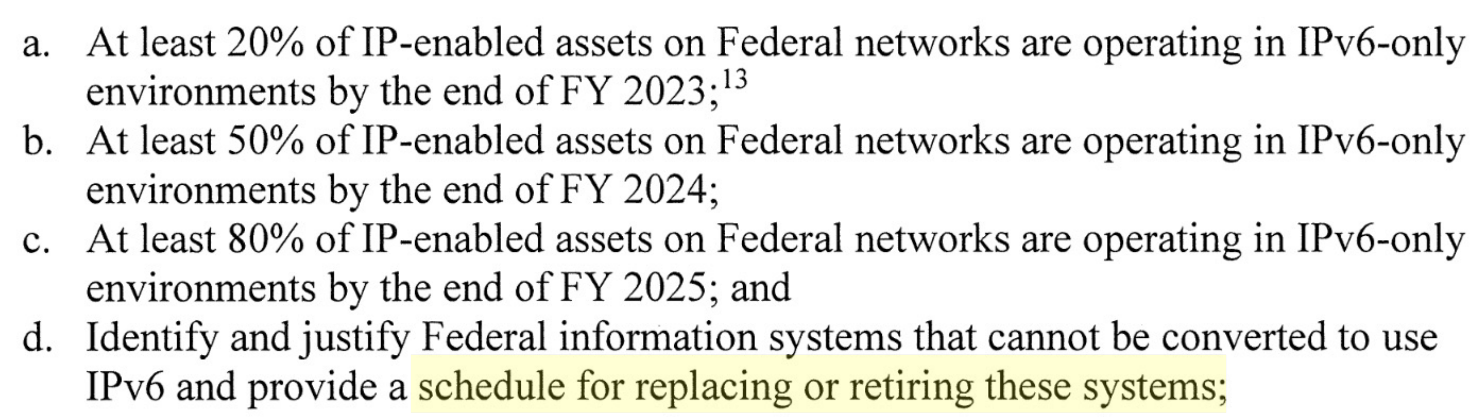
November 19, 2020

M-21-07

MEMORANDUM FOR HEADS OF EXECUTIVE DEPARTMENTS AND AGENCIES

FROM: Russell T. Vought
Director

SUBJECT: Completing the Transition to Internet Protocol Version 6 (IPv6)



a. At least 20% of IP-enabled assets on Federal networks are operating in IPv6-only environments by the end of FY 2023;[13]
b. At least 50% of IP-enabled assets on Federal networks are operating in IPv6-only environments by the end of FY 2024;
c. At least 80% of IP-enabled assets on Federal networks are operating in IPv6-only environments by the end of FY 2025; and
d. Identify and justify Federal information systems that cannot be converted to use IPv6 and provide a schedule for replacing or retiring these systems;

Basics

IPv4	IPv6
4-byte address	16-byte address and thus different IP header
Decimal notation *127.0.0.1,* *160.91.134.109*	Hex notation *0000:0000:0000:0000:0000:0000:0000:0001* = *::1* *fe80:0000:0000:0000:020c:29ff:feab:2e3c* = *fe80::20c:29ff:feab:2e3c*
.. With port: *127.0.0.1:5076*	.. Uses [ip]:port *[::1]:5076*

Good News

IPv4	IPv6
DNS gives IP4 address	DNS can also give IP6 address
gethostbyname("name" or "1.2.3.4") → *sockaddr_in*	*getaddrinfo("name" or "1.2.3.4" or "abcd:1234::1")* → *sockaddr_in* or *sockaddr_in6*
Socket API: *connect(), bind(), send(), recv(), select(), read(), write(), …, close()*	THE SAME!!
TCP	THE SAME!!

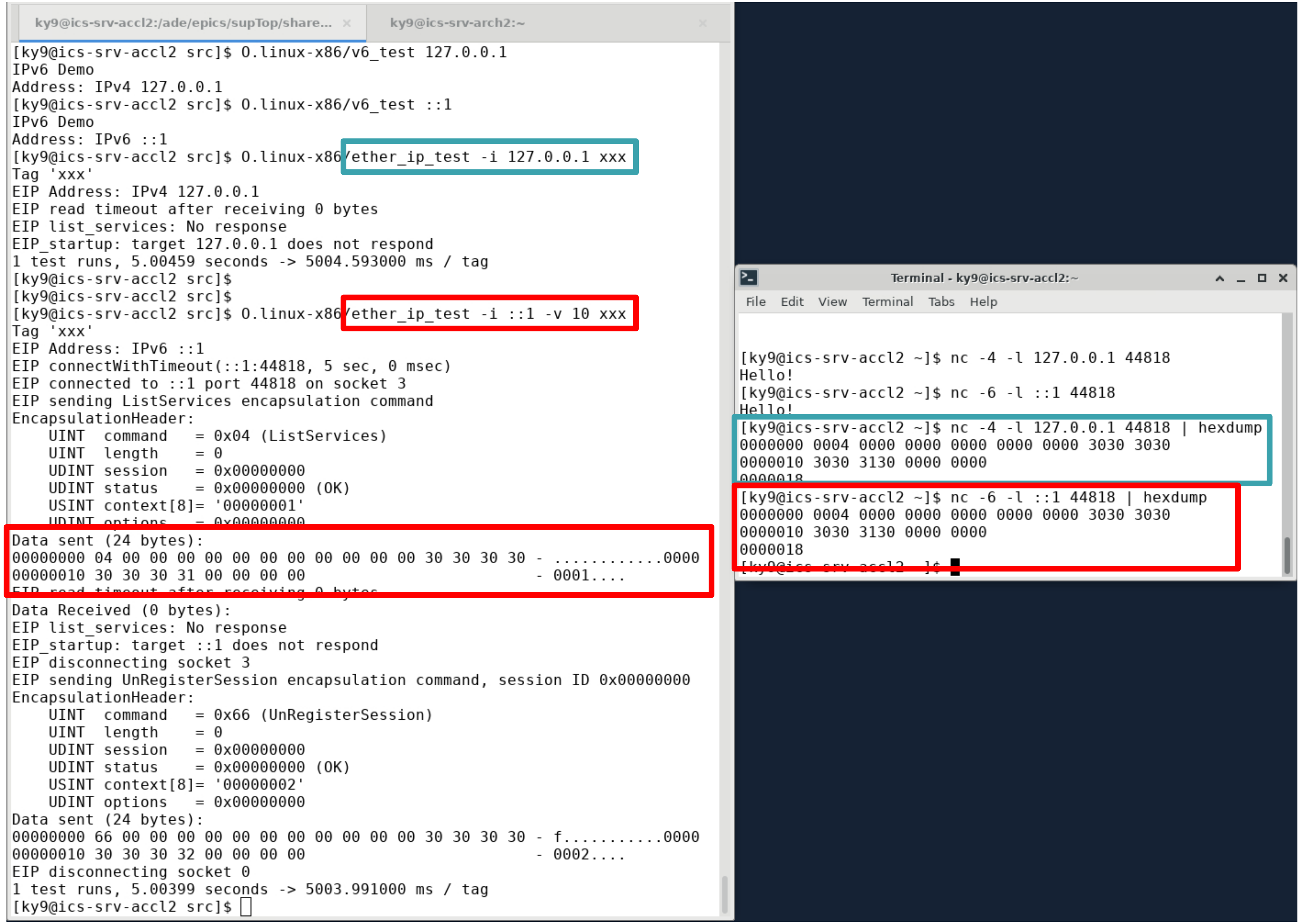
➔ Porting C/C++/Java/python/.. code can be as simple as using getaddrinfo()

EtherIP driver/device (Allen-Bradley Control Logix PLCs)

😎 Now supports IPv6

😢 No PLC to talk to, yet

We use “UCMM” mode via a TCP connection and can connect via IPv4 or IPv6, not passing any network addresses inside protocol messages. There are EtherNet/IP messages that embed network addresses, which are IPv4-only and will require protocol updates to support IPv6.


ky9@ics-srv-accl2:/ade/epics/supTop/share...
ky9@ics-srv-arch2:~
Terminal - ky9@ics-srv-accl2:~
File
Edit
View
Terminal
Tabs
Help

```
[ky9@ics-srv-accl2 src]$ O.linux-x86/v6_test 127.0.0.1
IPv6 Demo
Address: IPv4 127.0.0.1
[ky9@ics-srv-accl2 src]$ O.linux-x86/v6_test ::1
IPv6 Demo
Address: IPv6 ::1
[ky9@ics-srv-accl2 src]$ O.linux-x86/ether_ip_test -i 127.0.0.1 xxx
Tag 'xxx'
EIP Address: IPv4 127.0.0.1
EIP read timeout after receiving 0 bytes
EIP list_services: No response
EIP_startup: target 127.0.0.1 does not respond
1 test runs, 5.00459 seconds -> 5004.593000 ms / tag
[ky9@ics-srv-accl2 src]$
[ky9@ics-srv-accl2 src]$
[ky9@ics-srv-accl2 src]$ O.linux-x86/ether_ip_test -i ::1 -v 10 xxx
Tag 'xxx'
EIP Address: IPv6 ::1
EIP connectWithTimeout(::1:44818, 5 sec, 0 msec)
EIP connected to ::1 port 44818 on socket 3
EIP sending ListServices encapsulation command
EncapsulationHeader:
    UINT  command    = 0x04 (ListServices)
    UINT  length     = 0
    UDINT session    = 0x00000000
    UDINT status     = 0x00000000 (OK)
    USINT context[8]= '00000001'
    UDINT options    = 0x00000000
Data sent (24 bytes):
00000000 04 00 00 00 00 00 00 00 00 00 00 00 30 30 30 30 - ............0000
00000010 30 30 30 31 00 00 00 00                          - 0001....
EIP read timeout after receiving 0 bytes
Data Received (0 bytes):
EIP list_services: No response
EIP_startup: target ::1 does not respond
EIP disconnecting socket 3
EIP sending UnRegisterSession encapsulation command, session ID 0x00000000
EncapsulationHeader:
    UINT  command    = 0x66 (UnRegisterSession)
    UINT  length     = 0
    UDINT session    = 0x00000000
    UDINT status     = 0x00000000 (OK)
    USINT context[8]= '00000002'
    UDINT options    = 0x00000000
Data sent (24 bytes):
00000000 66 00 00 00 00 00 00 00 00 00 00 00 30 30 30 30 - f...........0000
00000010 30 30 30 32 00 00 00 00                          - 0002....
EIP disconnecting socket 0
1 test runs, 5.00399 seconds -> 5003.991000 ms / tag
[ky9@ics-srv-accl2 src]$
```

```
[ky9@ics-srv-accl2 ~]$ nc -4 -l 127.0.0.1 44818
Hello!
[ky9@ics-srv-accl2 ~]$ nc -6 -l ::1 44818
Hello!
[ky9@ics-srv-accl2 ~]$ nc -4 -l 127.0.0.1 44818 | hexdump
0000000 0004 0000 0000 0000 0000 0000 3030 3030
0000010 3030 3130 0000 0000
0000018
[ky9@ics-srv-accl2 ~]$ nc -6 -l ::1 44818 | hexdump
0000000 0004 0000 0000 0000 0000 0000 3030 3030
0000010 3030 3130 0000 0000
0000018
[ky9@ics-srv-accl2 ~]$
```

Details…

IPv4	IPv6
TCP IP addresses in PV Access protocol (search response, …) always reserved 16 bytes. Channel Access uses 4-byte addresses in protocol messages…	THE SAME!! Sockets in fact default to bilingual "tcp46" type. But beware: 1) Program A creates *tcp4* socket, binds to port 1234 2) Program B creates *tcp46* socket, binds to port 1234 → Succeeds! Program B believes it owns that port, but it will only receive IPv6 traffic. IPv4 goes to Program A → *PVA server tests tcp4 socket on port before creating tcp46 socket*
UDP: unicast, broadcast, multicast (seldom used?)	UDP: unicast, NO broadcast, multicast (allegedly improved with predefined multicast address ranges)

No Broadcast?

Channel Access and PV Access, on IPv4, broadcast to resolve PVs

- Unicasts only received by one listener
- Broadcast can reach multiple IOCs per host

Multicasts *ff..:* offer the same idea:

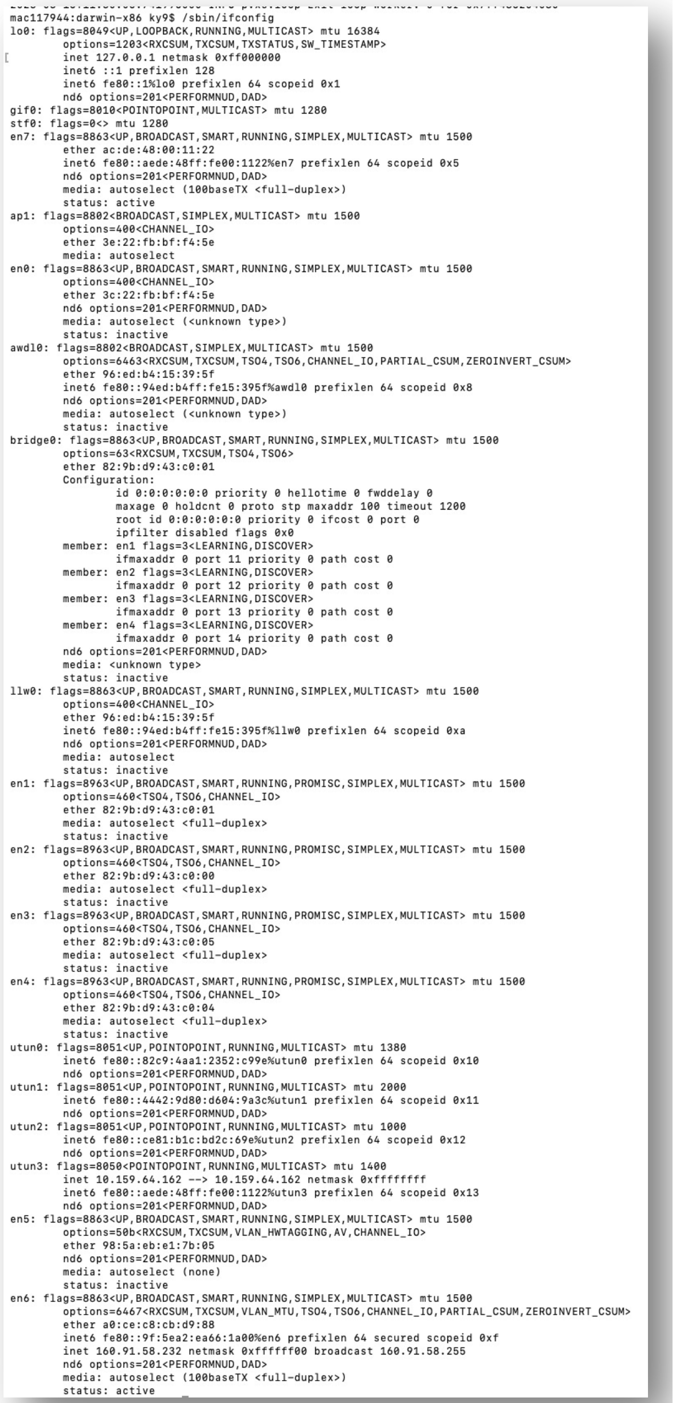
ff02:...	Link-local multicast
ff02::1	Link-local “all nodes” multicast, basically broadcast
ff02::42:1	Suggestion for PV Access Name Search multicast

IPv6 Address Caveats

- Many computers only have auto-assigned link-local addresses
  - Start with *fe80:…*, end with *%interface*
  - Example: *fe80::9f:5ea2:ea66:1a00%en6*

- Multicast address needs to be used with interface
  - *[ff02::42:1]@en6*

- `/sbin/ifconfig` can be long,
  hard to support EPICS_PVA_AUTO_ADDR_LIST



```
mac117944:darwin-x86 ky9$ /sbin/ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en7: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether ac:de:48:00:11:22
        inet6 fe80::aede:48ff:fe00:1122%en7 prefixlen 64 scopeid 0x5
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (100baseTX <full-duplex>)
        status: active
ap1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 3e:22:fb:bf:f4:5e
        media: autoselect
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 3c:22:fb:bf:f4:5e
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (<unknown type>)
        status: inactive
awdl0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
        options=6463<RXCSUM,TXCSUM,TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether 96:ed:b4:15:39:5f
        inet6 fe80::94ed:b4ff:fe15:395f%awdl0 prefixlen 64 scopeid 0x8
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (<unknown type>)
        status: inactive
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=63<RXCSUM,TXCSUM,TSO4,TSO6>
        ether 82:9b:d9:43:c0:01
        Configuration:
                id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
                maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
                root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
                ipfilter disabled flags 0x0
        member: en1 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 11 priority 0 path cost 0
        member: en2 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 12 priority 0 path cost 0
        member: en3 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 13 priority 0 path cost 0
        member: en4 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 14 priority 0 path cost 0
        nd6 options=201<PERFORMNUD,DAD>
        media: <unknown type>
        status: inactive
llw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 96:ed:b4:15:39:5f
        inet6 fe80::94ed:b4ff:fe15:395f%llw0 prefixlen 64 scopeid 0xa
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: inactive
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:9b:d9:43:c0:01
        media: autoselect <full-duplex>
        status: inactive
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:9b:d9:43:c0:00
        media: autoselect <full-duplex>
        status: inactive
en3: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:9b:d9:43:c0:05
        media: autoselect <full-duplex>
        status: inactive
en4: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        options=460<TSO4,TSO6,CHANNEL_IO>
        ether 82:9b:d9:43:c0:04
        media: autoselect <full-duplex>
        status: inactive
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
        inet6 fe80::82c9:4aa1:2352:c99e%utun0 prefixlen 64 scopeid 0x10
        nd6 options=201<PERFORMNUD,DAD>
utun1: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
        inet6 fe80::4442:9d80:d604:9a3c%utun1 prefixlen 64 scopeid 0x11
        nd6 options=201<PERFORMNUD,DAD>
utun2: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1000
        inet6 fe80::ce81:b1c:bd2c:69e%utun2 prefixlen 64 scopeid 0x12
        nd6 options=201<PERFORMNUD,DAD>
utun3: flags=8050<POINTOPOINT,RUNNING,MULTICAST> mtu 1400
        inet 10.159.64.162 --> 10.159.64.162 netmask 0xffffffff
        inet6 fe80::aede:48ff:fe00:1122%utun3 prefixlen 64 scopeid 0x13
        nd6 options=201<PERFORMNUD,DAD>
en5: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=50b<RXCSUM,TXCSUM,VLAN_HWTAGGING,AV,CHANNEL_IO>
        ether 98:5a:eb:e1:7b:05
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (none)
        status: inactive
en6: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6467<RXCSUM,TXCSUM,VLAN_MTU,TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether a0:ce:c8:cb:d9:88
        inet6 fe80::9f:5ea2:ea66:1a00%en6 prefixlen 64 secured scopeid 0xf
        inet 160.91.58.232 netmask 0xffffff00 broadcast 160.91.58.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (100baseTX <full-duplex>)
        status: active
```

Configuration

Server:

EPICS_PVAS_INTF_ADDR_LIST

“0.0.0.0 [::]”	All interfaces, IPv4 and IPv6
“[::]”	.. IPv6 only
“[::1]”	IPv6 localhost only
“[::] [ff02::42:1]@en6”	IPv6, any interface + Multicast

Client:

EPICS_PVA_AUTO_ADDR_LIST=NO

EPICS_PVA_ADDR_LIST

“[::1]”	IPv6 unicast to localhost
“[ff02::42:1]@en6”	Multicast
“[fe80::9f:5ea2:ea66:1a00]@en6”	Unicast to IP

TCP Name Resolution

- Server always responds to name searches via its TCP port

- Client needs to be configured to

  not use UDP

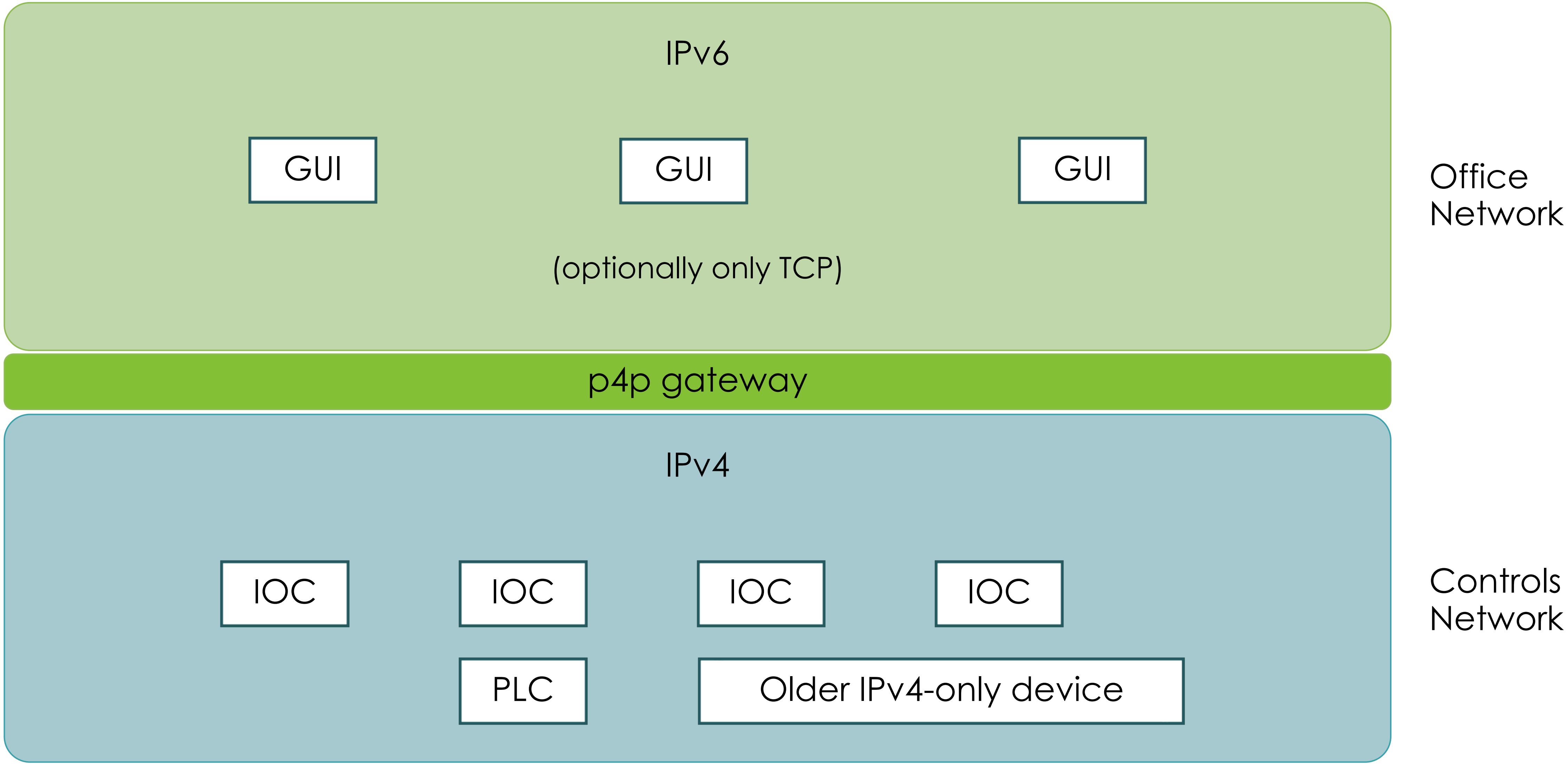
  - EPICS_PVA_AUTO_ADDR_LIST="NO"
  - EPICS_PVA_ADDR_LIST=“”

  and instead use TCP

  - EPICS_PVA_NAME_SERVERS="[::1]:5075”   for ”localhost” test
  - EPICS_PVA_NAME_SERVERS="[fe80::9f:5ea2:ea66:1a00]:5075”   for some host

This can be a good way to connect to the gateway!

Gateway can bridge IPv4 / IPv6



PV Access Implementations

C++: PVXS

- pvxget, pvxmonitor, …
- p4p python binding,
- p4p gateway

Java: core-pva

- CS-Studio (phoebus)
- Matlab

Not supported by older libraries

C++: pvAccessCPP

- EPICS base ‘qsrv’ ➔ *“PVXS in your IOC”, Michael Davidsaver*

Java: pvAccessJava

- CS-Studio (Eclipse)

Summary

Latest PV Access implementations support IPv6

✓ ”Multicasts” instead of “Broadcasts”,   *ff02::42:1@interface*

✓ Optional TCP-only mode

✓ Gateway can bridge IPv4 / IPv6

At this time,

☐ Little operational experience

☐ Mostly auto-assigned link-local IPv6 settings

☐ Best use EPICS_PVA_AUTO_ADDR_LIST=NO,
set EPICS_PVAS_INTF_ADDR_LIST
and EPICS_PVA_ADDR_LIST