TINE Release 4.0 News
(May 7, 2010: That was the month that was !)

"What a long, strange trip it's been …."

New Features: Security

- **Network Address** Access Restriction
  - via File (*in [FEC_HOME]/<EQM>*)
    - ipnets.csv
    - <property>-ipnets.csv
      - property-default-ipnets.csv
    - <device>-ipnets.csv
      - device-default-ipnets.csv
  - via API
    - AppendRegisteredNetsList(eqm, iplist, listsize)
    - AssignPropertyNetsList(eqm, prp, ipnets, nipnets)
    - AssignDeviceNetsList(eqm, prp, ipnets, nipnets)

Equivalent methods in java (TEquipmentModule) …

Address Lists …

SUBNET
131.169.9.255 → **All addresses on subnet .9.**

0.0.0.0 → **The local host via SHM**

131.169.120.47 → **The specific host @ 131.169.120.47**

And now :

SUBNET
131.169.150.255
131.169.9.64/26 → **131.169.9.64 -> 131.169.9.127**
0.0.0.0
131.169.119.0/26 → **131.169.119.0 -> 131.169.119.63**
131.169.120.0/24 → **131.169.120.0 -> 131.169.120.255**
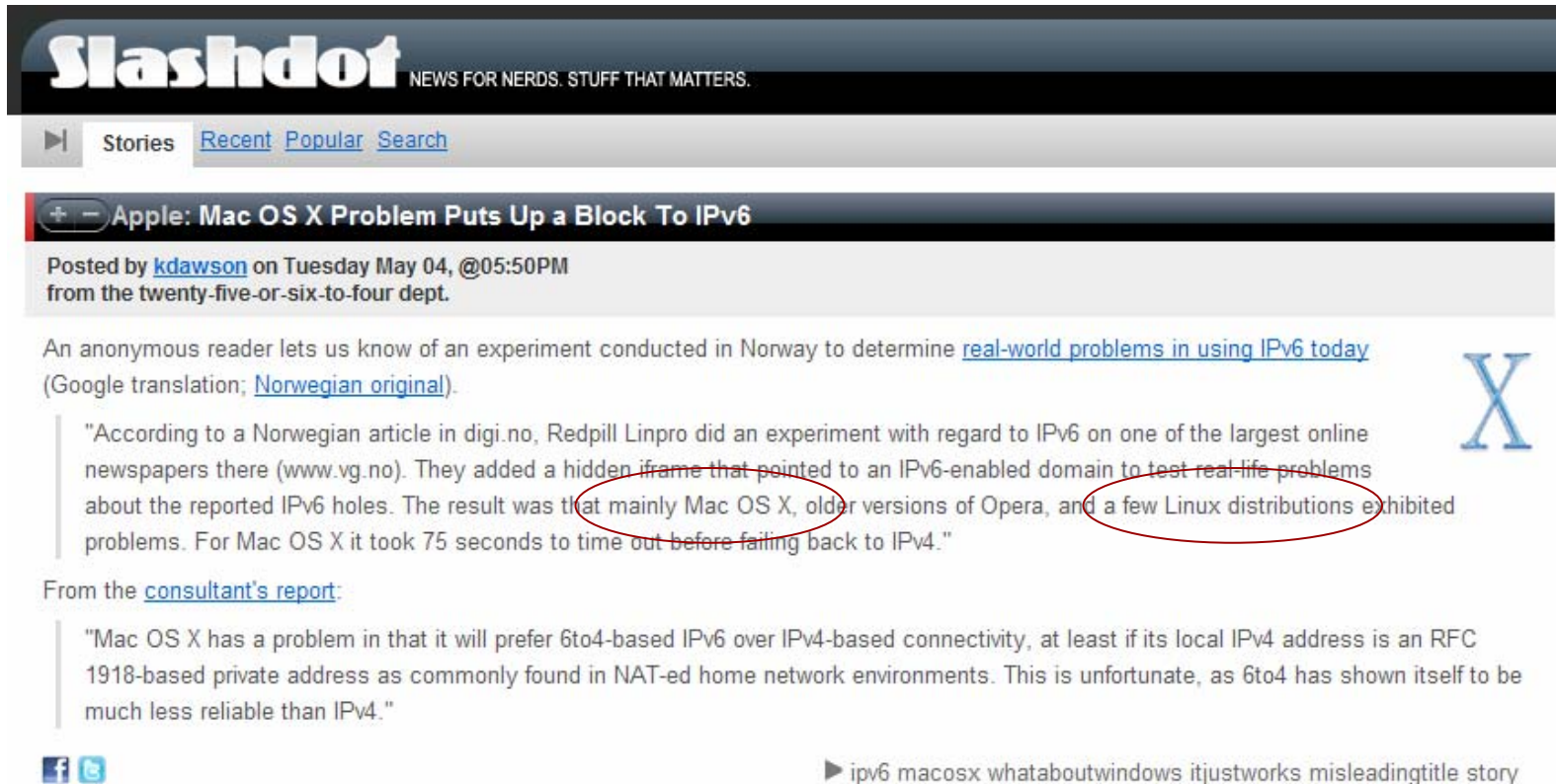
CIDR *(Class Inter Domain Routing)* notation

(equivalent to 131.169.120.255)

IPv4 versus IPv6

It just won't go away …

http://apple.slashdot.org/story/10/05/04/2029255/Mac-OS-X-Problem-Puts-Up-a-Block-To-IPv6

IPv4 versus IPv6 (work-around)

```java
public TPacket(int port,int rcvBufferSize,int timeToLive)
{
    int clnport = initializer.getClnPort();
    boolean useNextPort = false;
    boolean socketBound = false;
    InetAddress grp;
    for (int i=0; !socketBound && i<1000; i++)
    {
        try
        {
            if (port == 0)
            { // general client side port
                useNextPort = true;
                InetAddress iaddr = InetAddress.getLocalHost();
                if (os != null && os.contains("Linux")) iaddr = null;
                InetSocketAddress addr = new InetSocketAddress(iaddr,clnport+i);
                sck = new MulticastSocket(null);
                sck.setReuseAddress(false);
                sck.bind(addr);
                sck.setSoTimeout(0);
                sck.setReceiveBufferSize(rcvBufferSize);
                sck.setTimeToLive(timeToLive);
                socketBound = true;
                this.port = clnport + i;
            }
            else if (port == initializer.getGCastPort())
            { // globals listening port
                grp = InetAddress.getByName(initializer.getGCastAddress());
```

MAC OS needs this

Linux gets confused by it

Windows doesn't care

And Now …

- A Tale of Two Servers …

Magnet Server : PETRA

Magnet Server : FLASH

Servers under Attack …

- Recent Features to improve performance.
  - Prevent redundant 'contract list' scans !
    - Get/SetSchedulerInterval()
      - Default = 0 (as often as necessary)
      - Non-zero value -> minimum interval (but introduces latency!)
    - Get/SetSystemCycleDeadband()
      - Default ~ 10 msec (most OSes)
      - Gives time to wait for a server request
      - Blocks the scheduler until a request comes in (no latency problem!)
    - Distinguish between contract 'renewal' or 'cancel' AND 'new request'.'
      - 'deadband' wait broken by any of these, but pass thru scheduler blocked if nothing 'new'!

Servers under Attack …

- How to coerce 'single-device' panel or MatLab applications into efficient data access ?
  - Avoid synchronous polling !
    - The server could 'enforce' this (but that's rather harsh).
    - Make MatLab, etc. use asynchronous 'listeners' (i.e. synchronous calls from a local buffer).
  - Enforce Multi-Channel Array (MCA) access
    - Need to register properties as such !
      - OO issues (Karol is working on this)
    - DOOCS enquires as to data type/size need to 'see' a 'single value' as the data size
      - DDD panels don't know how to deal with a MCA as is!
      - CF_DEFAULT will report a size of '1' if property is a registered MCA -> single-element link logic is already in place !
      - If 'rpc_test' sees a MCA property then show the user a '1' as the data size!
  - If Necessary: enforce NETWORK (i.e. multicast) access

Feature-of-the-Month AWARD

MCS–1

– (Room 520) For dealing with this!

Plus: a bronze star as well :

doocs2tine News

- Current deployed version contains the necessary client-logic to 'react' to a 'CF_DEFAULT' request to a MCA property properly.
  - Next TINE server release will now fix CF_DEFAULT requests for MCA properties to use a size of '1' (the DOOCS preference).

Multi-Channel Arrays (a short review)

See http://adweb.desy.de/mcs/tine/tinearraytype.html

- If you register a property as Multi-Channel Array then it should behave like one! (this is a 'covenant')
  - If property "P" is registered MCA then:
    - "P" will provide an array of length N (same EGU and settings)
    - Each element of "P" corresponds to a device channel
    - The order of the device channels in the array can be obtained by calling "P.NAM"
      - Various ways to achieve this
        - explicit property registration
        - Device assignment via API or file
        - Default to the registered DEVICES list
    - The Device specified in the call to "P" gives the desired 'offset' in the MCA.
    - The Array Length specified in the call gives the number of elements desired (usually '1' or 'N').
    - The behavior of the call when (requested length) > (N – offset) is left unspecified, but the server should NOT return an error.
      - One possibility: Wrap the array
      - Another possibility: Truncate the call

Multi-Channel Arrays (a short review)

- Single-element access of MCA property "P" flows thru internal handshaking
  - returns the entire array to the caller
  - informs him which element is desired!
- Application asking for all (or many) channels individually
  - collapses to a single contract
  - single interrupt at the server
- Major efficiency advantage
- Numerous use case 'bugs' fixed over past several months.
  - Most recent:
    - Single-element call with a non-existent device
    - Single-element call with an expired synchronous contract still in link table.
  - Other 'cute' use cases:
    - Multiple identical, single-element links using CF_DEFAULT and landing on a redirected property …

Multi-Channel Arrays (a short review)

- You can turn the 'handshaking' OFF (if desired):
  - SetEnforceMultiChannelAccess(TRUE or FALSE) (Default = TRUE)
- Advantages
  - Much more efficient
    - 1 dispatch interrupt instead of many
    - Less network traffic
      - Overhead of single contract instead of many
  - Atomic data access
    - Some 'collections' need to be atomic !
  - Archiving is much, much easier
    - Easier to configure
    - Easier to retrieve
    - 'Snapshots' of MCA at a given time
- This is a 'property' oriented view
  - Similar but NOT equivalent to wildcard '*' calls
    - With an MCA you know a priori what will come back
    - With a wildcard call you don't!

A working example :

Has 14 links instead of ~110!

New Stock Properties

- "SRVSETTINGS"

Same information as at command line:

New Stock Property Behavior

- **"SRVINIT"**
  - No changes
  - calls an equipment module's registered init routine
- **"SRVEXIT"**
  - No changes
  - Takes exit level as input
  - Causes a 'clean' exit of the server
- **"SRVRESET"**

  Not yet available in java!
  - *New behavior!*
  - De-registers all equipment modules (calls their exit routines)
  - Returns all allocated (tine) memory to the heap
  - Calls 'PostSystemInit()' (IF registered)
  - Can be used to effectively 'warm' restart the server!

Note: Remote Management MUST be enabled !

SetAllowRemoteManagement(TRUE)

New Registration Routines

- C-Lib only:
  - SetPreSystemInitFunction(), SetPostSystemInitFunction()

If "SRVRESET" is supposed to actually work properly !

Notable Bug Fix

- In the buffered 'listener'
  - Note: this is a utility add-on (tbufsrv.dll/.so)
    - Used in LabView, MatLab
    - And tine2tango (soon) !
  - Links with 'adjustable length data types' not working properly !
    - CF_STRING (free string)
    - CF_AIMAGE (adjustable image)
    - CF_ASPECTRUM (adjustable spectrum)

e.g. aIsnExecLink("/PETRA/Undulator/PU03","ConnectionErrorString", …)

requires data type CF_STRING !

CDI News

- Note: CDI devices with 'decorated' names
  - e.g. "BL_0.TEMP_ALARM_OUT"
  - Available as is via the "bus action" properties
    - "RECV", "SEND", "RECV.CLBR", etc.
  - Also 'parsed' and offered as 'extended' (multi-channel array) properties
    - Property "TEMP_ALARM_OUT" with an associated device list:
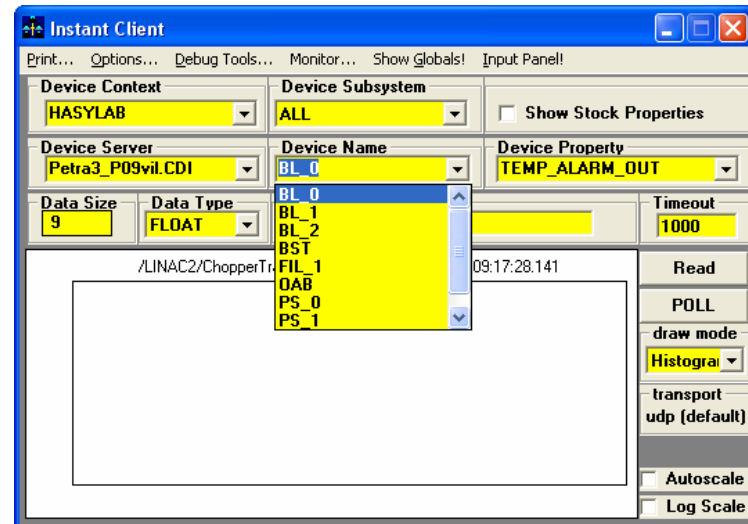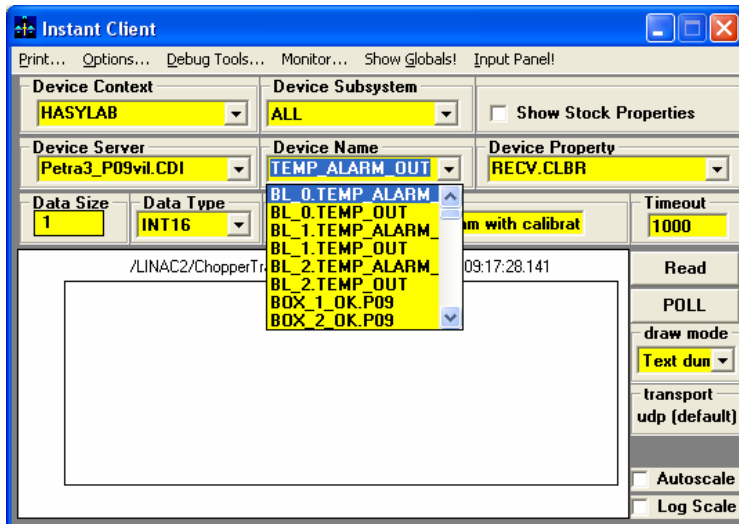
CDI News

- But, by default, CDI servers call:
  - *SetEnforceMultiChannelAccess*(FALSE)
  - i.e. assume that the device server programmer knows best how to read out his hardware efficiently !

- However, some CDI servers are used directly as device servers !

CDI News

- Can override the CDI default setting by
  - API: cdiSetEnforceMcaAccess(TRUE)
- Or (Ruediger's favorite)
  - set CDI_FORCE_MCA_ACCESS=TRUE
- If 'TRUE' then the single-element 'handshaking' is active !
  - But direct access from client applications should use the 'extended' properties

CDI News (example)

This is NOT a moot point: each contract results in an ADS access over EtherCat !

Video System News

- Lots of news, but have to wait until next time !