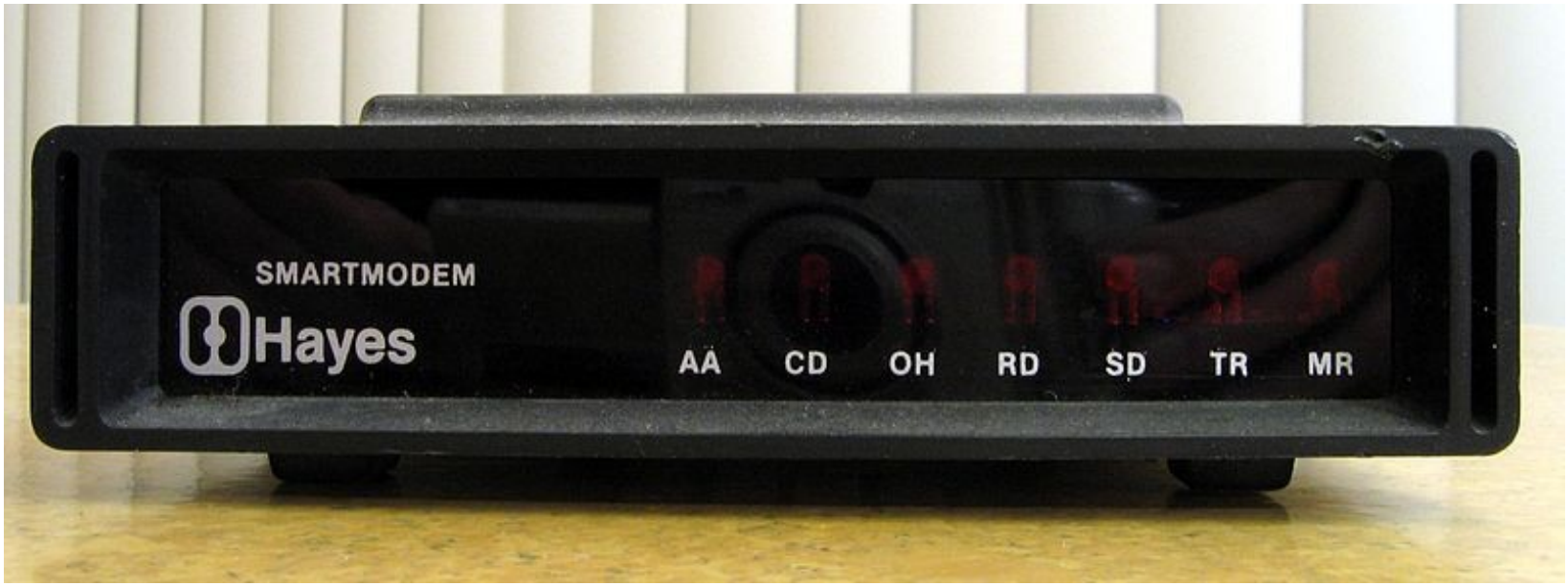
Lanedo

So your mobile broadband modem speaks... **what?**

Aleksander Morgado <aleksander@lanedo.com>

A long time ago in a galaxy not so far away...



Hayes AT command set

- 1981... !!
- Dial-up modems
- RS232

Hayes AT command set

- Command mode vs Data mode
  - ATDT15551234
  - ....
  - +++ATH
- PPP as data link session between modems
- Purpose:
  - Call management
  - Serial line setup (e.g. Flow control)

Hayes AT command set

- Different 'Hayes-compatible' vendors used the command set, lots of them in a completely **incompatible** way.

- 1995: V.25 series (ITU-T)
  - Attempt to establish a standard for the command set.

/r/gonemobile

Are you over 18 to see **mobile** content?

(c) Dan Williams, http://blogs.gnome.org/dcbw/

/r/gonemobile

- [1] Wireless communications
- [2] USB: multiple serial ports
- [3] USB: 'net' ports
- [4] Non-AT protocols
  - [4.1] QCDM
  - [4.2] WMC
  - [4.3] QMI
  - [4.4] MBIM
  - [4.5] Other

[1] Wireless communication

- Not only that a cable is missing between modems, **new features** as well:
  - Signal quality
  - Access technology
  - SMS
  - USSD
  - SIM cards

[1] Wireless communication

- 3GPP2, Iridium:
  - AT commands to talk to the modem
  - Dial-up connections between two modems.
  - PPP when in data mode
    - PPP session between host and operator network.
    - CDMA, EV-DO (ATDT#777)
    - Iridium (ATDT008816000025)

[1] Wireless communication

- 3GPP:
  - AT command set is extended with GPRS specific commands
  - ETSI GSM 07.07 / **3GPP TS 27.007**
  - Connection setup by activating user-defined or pre-defined PDP contexts.
  - PPP when in data mode
    - PPP only between host and local modem!
    - ATD*99

[1] Wireless communication

- 3GPP technical specifications **aren't enough**
  - Mode selection (e.g. 2G-only, 3G preferred, ...)
    - e.g. AT+ZSNT (ZTE)
  - Frequency band selection
    - e.g. AT^SCFG (Cinterion)
  - SIM state reporting
    - e.g. ^SIMST (Huawei)
  - Access technology reporting
    - e.g. AT_OWCTI, AT_OSSYS (Option)

[2] USB: multiple serial ports

- Mobile modems are no longer RS232 only
- Modems may expose **multiple** serial ports
- Modems can finally have status updates (e.g. signal quality) **while** connected
- GPS-enabled modems
  - New AT commands to setup GPS
    - e.g, AT_OGPS (Option)

[3] USB: 'net' interfaces

- PPP is dead, long live **net** interfaces
- ECM: Ethernet Control Model
  - USB full speed devices (e.g. cable modems)
  - 802.3 ethernet frames
- NCM: Network Control Model
  - Adjusts data transfer protocol, much more efficient.
  - Targeted for high-speed networks like HSPA or LTE
  - But not really suitable for mobile broadband
    - 802.3 ethernet frames don't really apply

[3] USB: 'net' interfaces

- Either with a custom kernel driver...
  - Sierra:
    - 'sierra_net' kernel driver
    - AT!SCACT
  - High-speed Option:
    - 'hso' kernel driver
    - AT_OWANCALL
- ...or with the standard ECM/NCM drivers:
  - Huawei
    - AT^NDISDUP

[4.1] Non-AT protocols: QCDM

- Qualcomm's **QCDM/Diag**
  - Binary protocol, proprietary.
  - Control protocol over serial port.
  - Data over serial port (PPP)
  - Primarily for status updates and diagnostics while connected.
  - ModemManager/libqcdm

[4.2] Non-AT protocols: WMC

- Pantech's **WMC**
  - Binary protocol, proprietary.
  - Control protocol over serial port.
  - Data over 'net' interface.
  - Call management, status updates...
  - ModemManager/libwmc
    - Not used in ModemManager yet, though.

[4.3] Non-AT protocols: QMI

- Qualcomm's **QMI**
  - New '**qmi_wwan**' kernel driver in upstream Linux 3.4
  - Modems expose a /dev/cdc-wdm port for QMI.
    - /dev/cdc-wdm is **NOT** serial!
  - Binary protocol, proprietary.
  - Data over 'net' port

[4.3] Non-AT protocols: QMI

- Protocol architecture:
  - Services
    - e.g. CTL, DMS, NAS, WDS...
  - Multiple clients over the same port
    - Useful in the original QCUSBNet
  - Requests, Responses, Indications
  - TLVs defined in each message type
    - Integers, strings, arrays, structs...

[4.3] Non-AT protocols: QMI

- **libqmi**
  - qmicli
  - ModemManager >= 0.7
  - GObject introspection
    - (Not fully ready yet)
    - python, JS, ...
- Ofono

[4.3] Non-AT protocols: QMI

```
// ***************************************************************************
{  "name"    : "Get Capabilities",
   "type"    : "Message",
   "service" : "DMS",
   "id"      : "0x0020",
   "version" : "1.0",
   "output"  : [  { "common-ref" : "Operation Result" },
                  { "name"      : "Info",
                    "id"        : "0x01",
                    "mandatory" : "yes",
                    "type"      : "TLV",
                    "format"    : "sequence",
                    "contents"  : [ { "name"   : "Max Tx Channel Rate",
                                      "format" : "guint32" },
                                    { "name"   : "Max Rx Channel Rate",
                                      "format" : "guint32" },
                                    { "name"          : "Data Service Capability",
                                      "format"        : "guint8",
                                      "public-format" : "QmiDmsDataServiceCapability" },
                                    { "name"          : "SIM Capability",
                                      "format"        : "guint8",
                                      "public-format" : "QmiDmsSimCapability" },
                                    { "name"   : "Radio Interface List",
                                      "format" : "array",
                                      "array-element" : { "format"        : "guint8",
                                                          "public-format" : "QmiDmsRadioInterface" } } ],
                    "prerequisites": [ { "common-ref" : "Success" } ] } ] },
```

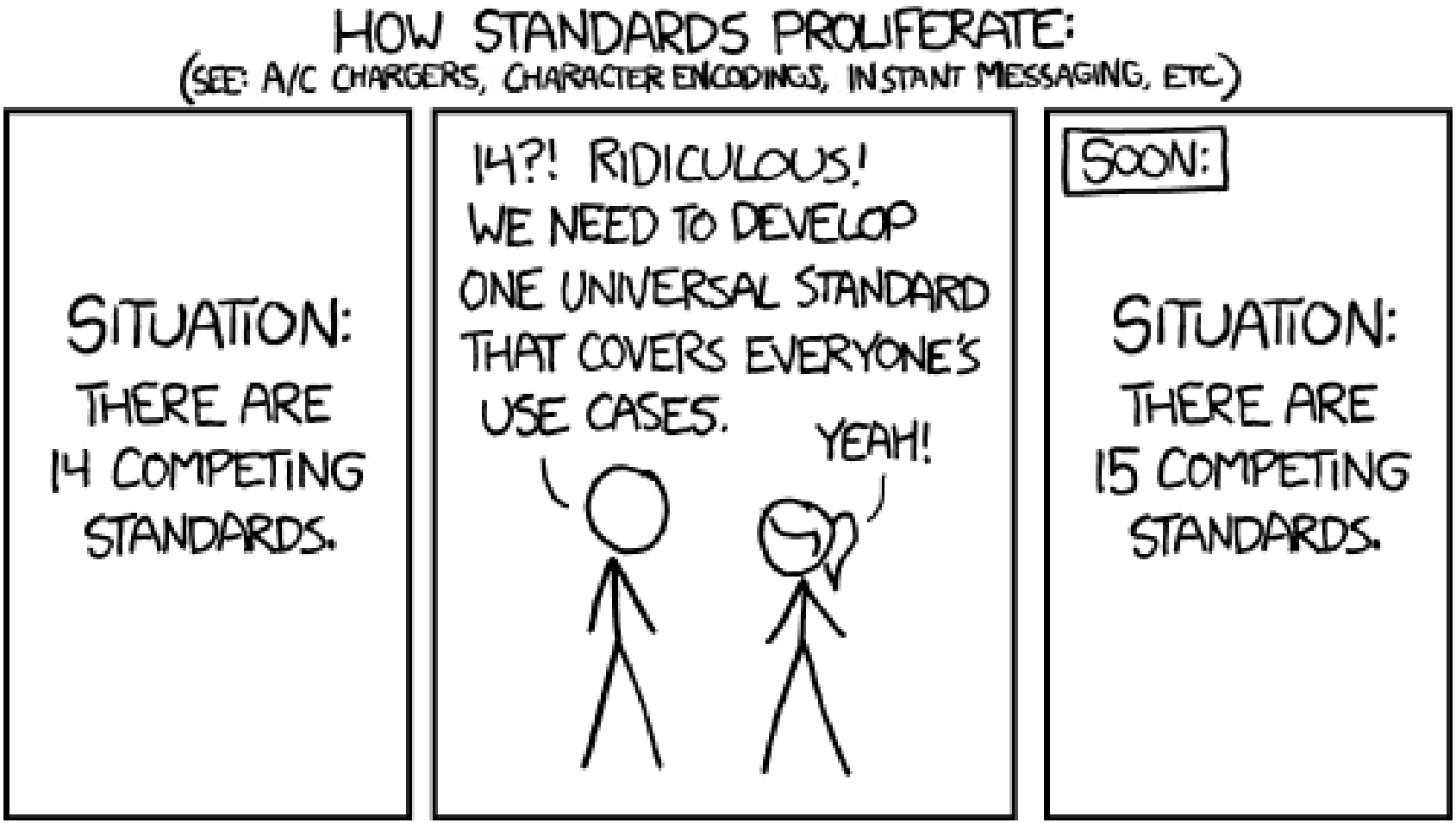
[4.3] Non-AT protocols: QMI

- qmicli allows performing QMI request+responses directly from the command line interface:

```
$ sudo qmicli \
         -d /dev/cdc-wdm0 \
         --dms-get-capabilities

[/dev/cdc-wdm0] Device capabilities retrieved:
Max TX channel rate: '5742000'
Max RX channel rate: '7200000'
       Data Service: 'non-simultaneous-cs-ps'
                SIM: 'supported'
           Networks: 'gsm, umts'
```


I WANT YOU
TO USE
MBIM
www.says-it.com/unclesam/



http://xkcd.com/927

[4.4] Non-AT protocols: MBIM

- Developed by the **USB IF**
  - Microsoft, Ericsson, Qualcomm...
- ECM->NCM->MBIM
  - Raw IP packets instead of Ethernet frames
  - Modem control protocol

[4.4] Non-AT protocols: MBIM

- New '**cdc_mbim**' kernel driver in upstream Linux 3.8
- Modems expose a /dev/cdc-wdm port for MBIM.
  - /dev/cdc-wdm is **NOT** serial!
- Binary protocol.
- Data over 'net' port

[4.4] Non-AT protocols: MBIM

- Protocol architecture:
  - Services
    - e.g. “Basic Connect”, “SMS”, “USSD”...
  - Requests, Responses, Indications
  - Fixed fields in each message:
    - Integers, strings, arrays, structs...
  - Message fragmentation
  - Other protocols may be embedded within MBIM
    - e.g. QMI inside MBIM

[4.4] Non-AT protocols: MBIM

- **libmbim**
  - mbimcli
  - ModemManager >= 0.7
    - (Not yet)
  - GObject introspection
    - (Not yet)
    - python, JS, ...

[4.4] Non-AT protocols: MBIM

```
// ***************************************************************************
{ "name"    : "Subscriber Ready Status",
  "service" : "Basic Connect",
  "type"    : "Command",
  "query"   : { "output" : [ { "name"          : "Ready State",
                               "format"        : "guint32",
                               "public-format" : "MbimSubscriberReadyState" },
                             { "name"          : "Subscriber ID",
                               "format"        : "string" },
                             { "name"          : "SIM ICCID",
                               "format"        : "string" },
                             { "name"          : "Ready Info",
                               "format"        : "guint32",
                               "public-format" : "MbimReadyInfoFlag" },
                             { "name"          : "Telephone Numbers Count",
                               "format"        : "guint32",
                               "visibility"    : "private" },
                             { "name"            : "Telephone Numbers",
                               "format"          : "string-array",
                               "array-size-field" : "Telephone Numbers Count" } ] } },
```

[4.4] Non-AT protocols: MBIM

- mbimcli allows performing MBIM request+responses directly from the command line interface:

```
$ sudo mbimcli \
        -d /dev/cdc-wdm1 \
        --basic-connect-query-subscriber-ready-status

[/dev/cdc-wdm1] Subscriber ready status retrieved:
      Ready state: 'device-locked'
    Subscriber ID: 'unknown'
        SIM ICCID: '98431031152008695 0F1'
       Ready info: 'unknown'
Telephone numbers: ''
```

[4.5] Non-AT protocols: other

- Sierra Wireless' CnS
- Nokia's phonet
- ...

Thanks! Questions?

More info...

+Aleksander Morgado
aleksander@lanedo.com
aleksander@gnu.org
https://sigquit.wordpress.com
http://www.lanedo.com

- **http://cgit.freedesktop.org/ModemManager/ModemManager**
- **http://cgit.freedesktop.org/libqmi**
- **https://gitorious.org/lanedo/libmbim (soon in fd.o hopefully)**