Standardizing LTE modems for IoT
Terry Moore, MCCI Corporation, USB-IF MBIM working group chair
MCCI Facts

- Founded in 1995
- An all-encompassing partner for USB system development
- Successfully distributed with more than a billion embedded products to date
- Experts in delivering advanced connectivity System Software for Enterprises
- Global expert support and project execution teams
- Advanced Connectivity Stacks for mobile broadband SoCs, Automotive Smartphone Connectivity and PC wireless docking
Standards and LTE modems
The need for standards in LTE modems

IoT / M2M “market” is really a collection of many markets

Vendors with specific vertical-market expertise want to add connectivity to their products
- Typically small to medium-size companies
- Or small vertical groups inside larger conglomerates

Many IoT applications have long installed lives (up to 30 years!)
- Networks and components evolve on a faster cycle.
- If modems are not standardized, then in five years there may be no way to replace modems, ship new systems, adapt to new networks, etc.
The need for standards in LTE modems

Vendors need standards in order to use 4G or 5G networks

- They won’t be able to design their own modems.
- For exporting, they’ll need to buy modems from the appropriate vendor for the target market (Chinese modem for China, Korean modem for Korea, etc.).
- Something has to be done to get their apps or equipment connected to the network.
- The value proposition of cellular-based approaches is that the operator takes care of all the infrastructure, for a fee.
- But if vendors have to customize their applications for every different modem, it will be a great burden for them.
- They will look to using Wi-Fi or other free standards for connectivity instead of 4G or 5G.
What needs to be standardized

Physical size of modem (PCIe minicard or USB stick)
Power and cooling requirements of modem (PCIe minicard or USB)
Antenna connections (PCIe minicard; built in for USB)
Bus interface of modem (USB)
Command set of modem (MBIM)
Network requirements for minimal commands needed to be sent to modem (de-facto standard set by Microsoft for MBIM)
Provisioning (the operators)

⇒ Standards exist, or are emerging, in each area
Who will drive standards?

Interesting question – not the usual suspects!

• The very largest modem users – Apple, Samsung, etc., have no need for standards, because they can design their own modems.

• The IoT vendors need standards, because they can’t afford to design modems. But if no standard exists, they’ll use other approaches (Wi-Fi, etc.) (And they can’t afford to help develop standards)

• The modem chipset makers (Qualcomm, Intel, etc.) already are supporting the relevant standards (MBIM, PCIe minicard)

• Sophisticated system software vendors (Microsoft, etc) are supporting MBIM

• The network operators should strongly support the standards effort, in order to enable the growth of IoT and embedded use cases for their networks.
Wi-Fi vs LTE

Wi-Fi connectivity is simpler (for a station)
Wi-Fi higher level software is widely deployed in Windows, Linux, ... so the hardware-specific driver can be pretty simple
Wi-Fi drivers are tied to specific hardware, but one hardware will work worldwide

LTE is inherently more complex (but better)
Embedded connection managers are not widely deployed in Windows, Linux, RTOS, so companies have to use something specific to a given LTE modem (or use MBIM modem)
LTE modems typically cannot be used world-wide (because you have to use what the operator permits)
What about 3GPP AT commands?

Problem is that 3GPP AT commands assume “serial port”

USB vendors use a number of different ways to connect

- RNDIS
- QMI
- ECM
- NCM
- CDC ACM
- Vendor-specific

So even if the commands are standardized, the transport (virtual serial port) is not standardized

And in fact, outside Japan, the AT commands are not at all standardized

Big headache for IoT equipment makers
MBIM Overview

MBIM stands for “Mobile Broadband Interface Model”

It’s a “USB class specification”

Defines the control plane and the data plane with enough detail to allow all the system software on the PC or IoT system to be written in a way that is portable from modem to modem

- Standardized ways of representing modem capabilities to the host
- Standardized commands and responses
- Standardized, very efficient data transport

Data is moved as IP datagrams (no more PPP, no more Ethernet emulation)

Commands are moved as binary messages, and can be multiplexed (no more AT commands)

Standardized ways of providing extensions (yes, can even support legacy AT commands)

Developed and licensed by USB-IF on a RAND-Z basis

Vigorously supported by Microsoft
Current Status

MBIM has been successful in the PC area
There now is MBIM support for Windows, Linux and MacOS from a variety of sources.

- Operators are starting to use MBIM for Windows 7 on their own initiative

Software vendors and modem vendors report reduced support costs. (Qualcomm, Smith Micro, MCCI reported at PCCA meeting last year)

MCCI is getting design wins from companies who want to embed MBIM support in embedded systems
Current participants

Ericsson
Intel
MCCI
Microsoft
Qualcomm
Smith Micro Software, Inc.
Synopsys
Next Steps

Modem makers have discovered a generic problem with USB modems when supporting multiple applications concurrently on the application processor (e.g. VoLTE plus file transfer).

A work item has been approved by USB-IF to address this problem with MBIM, “MBIM MultiFlow”.

The MBIM standards group needs carrier participation in order to be sure that the problem is solved in a way that matches network requirements.
MBIM MultiFlow

MBIM (like most USB transports) multiplexes all uplink traffic over a single bulk OUT pipe. The modem has some number of buffers for receiving MBIM NTBs (which potentially contain multiple IP frames)

If buffers get full, the only way for the modem to slow down traffic coming into the modem is by NACKing on the bulk pipe

In advanced applications, this can cause a number of problems similar to the router “buffer bloat” problem (http://en.wikipedia.org/wiki/Bufferbloat) problem

- Excessive jitter
- Excessive latency
- DNS timeouts

(But different, because of the characteristics of the USB transport)
A simple example

Alice places an VoLTE call to Bob, using the PC headset and microphone
- Audio data is digitized on the PC and flows over IP to the network
- The modem characterizes these IP packets as meeting some QoS profile, and the network grants higher QoS windows sufficient to keep the data flowing

While talking to Bob, Alice sends a large file to Bob
- This is sent with TCP, and output from the file transfer app to the IP stack quickly ramps up to keep the network link saturated

Then network congestion occurs
- The file transfer application continues sending data, but data cannot get to the network, because the network doesn’t offer sufficient “best effort” transmit opportunities, so the modem buffers fill.
- The modem starts NAKing on the OUT pipe

Subsequent audio traffic from Alice can’t get to the modem (because of the USB NAKs). This situation persists until either the modem discards traffic, or until the network moves enough best-effort traffic to drain the buffers on the modem
- This causes jitter / delay / bad audio at the receiver

The MultiFlow group is tasked to modify the MBIM spec in some way, to allow modem and host to cooperate to avoid this situation
Police Car example

Police car with multiple streams, some QoS

As with the Alice / Bob example, key factors are:

- QoS traffic
- Best effort traffic using TCP for uploads concurrent with QoS
- A downward fluctuation in network capacity for best-effort after TCP has ramped up the speed to match the old capacity

Police car generating the following traffic:

- QoS traffic for live video feed to dispatcher
- Best effort traffic for uploading information (accident photographs, reports, etc)
- Something bad happens nearby, and bystanders (or other responders) start loading the net
- live video feed degrades because modem buffers get full of best-effort traffic
Surveillance Camera

Surveillance camera with store/forward + live

As with the Alice / Bob example, key factors are:

- QoS traffic
- Best effort traffic using TCP for uploads concurrent with QoS
- A downward fluctuation in network capacity for best-effort after TCP has ramped up the speed to match the old capacity

Detailed scenario

- Remote area, multiple cameras, single terminal
- QoS traffic for a live feed from one camera
- Best effort traffic for uploading saved images from other cameras
- Best effort traffic for controlling the cameras
- Something happens to network capacity (negative fluctuation)
- live video feed degrades, can’t remotely control the camera (because ACKs can’t get upstream)
Conclusions

Modem standards are critical for successful use of LTE in embedded systems

MBIM addresses a key area of modem standardization (the control and data formats)

The first version of MBIM is being successfully deployed, and is sufficient for current needs

The MBIM standard activity enables industry collaboration to address emerging issues for applications sharing a single modem (VoLTE plus best effort)
If you want to participate

Join USB-IF (this costs $4,000/year)
Sign the MBIM IP agreement (basically a RAND-Z agreement)
Participate (occasionally or consistently, as you have resources) in the group discussions
  - Weekly telephone meetings
  - Email and document review
Thank You!
For more information: Terry Moore tmm@mcci.com