RTT TECHNOLOGY TOPIC
January 2002

3G Software

In previous Hot Topics we have discussed the impact of **handset hardware** on 'offered traffic', how CMOS imaging and MP4 encoders determine radio and network bandwidth requirements and the 'qualities' needed from the physical layer - dynamic range (bursty bandwidth) and control of end to end delay and delay variability.

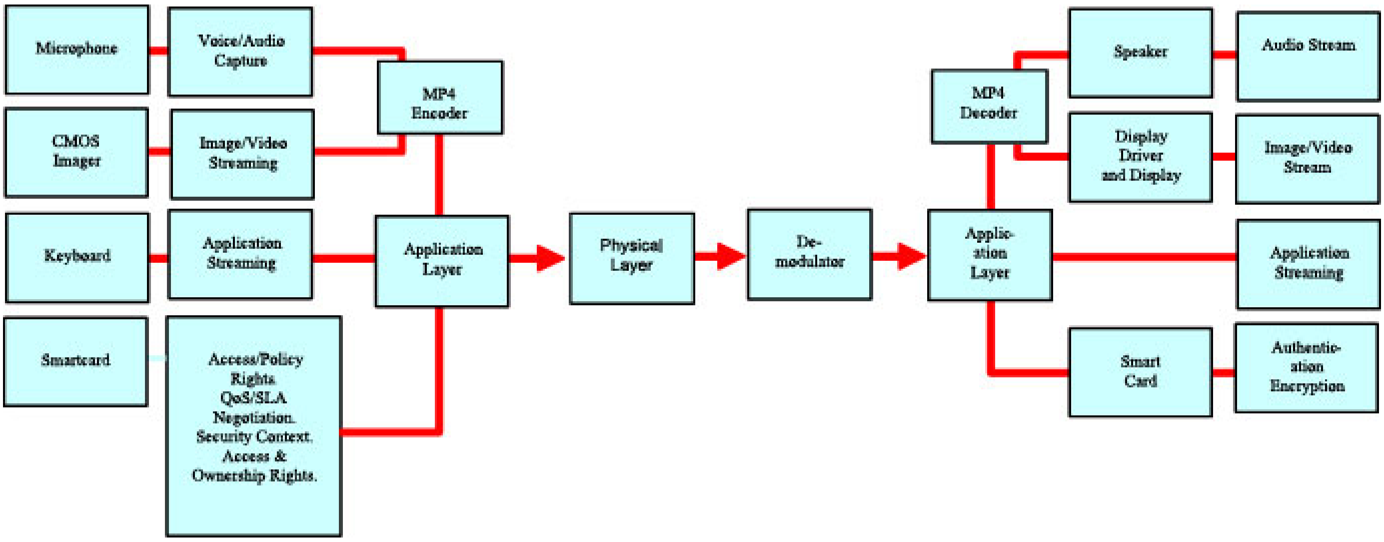

Figure 1 - Handset Hardware Form Factor and Functionality

Figure 1 shows a typical 3G handset with a microphone (audio capture!), CMOS imager and MP4 encoder (for image and video encoding), a keyboard (application capture), a smart card for establishing access and policy rights and (on the receive side) a speaker, display driver and display.

The addition of these **hardware components** (CMOS imager, MP4 encoder and high definition colour display) changes what a user can do and what a user expects from the device and from the network to which the device is connected.

This month we discuss how 3G **handset software** changes user behaviour and user expectations.

In all HOT TOPICS we generically refer to 3G. In Europe and Asia we tend to think of this as IMT2000DS but the comments are equally applicable to CDMA2000 1XRTT and 1XEV networks.

Note also, that in general, we are taking a three to five year (rather than three to five month) view of handset and network hardware and software design trends. Our aim is to show how a **long term** (three to five year) projection of technology and market transition can be used to realise

short term (three to five month) technology and market advantage.

Cellular Phone Generation	Memory Bandwidth	Processor Bandwidth(MIPS - millions of instructions per second)	Code Bandwidth (Lines of Code)
1G (1980's)	Kilobytes	10	10,000
2G (1990's)	Megabytes	100	100,000
3G (2000 - 2010)	Gigabytes	1000	1,000,000

Figure 2 -Software Footprint and Software Functionality

Software footprint and software functionality is a product of memory bandwidth (code and application storage space), processor bandwidth (the speed at which instructions can be processed) and code bandwidth (number of lines of code).

Over the past three generations of cellular phone, memory bandwidth has increased from (a few) kilobytes to (a few) megabytes to (a few) gigabytes, processor bandwidth has increased from 10 MIPS to 100 MIPS to 1000 MIPS and code bandwidth has increased from 10,000 to 100,000 to 1,000,000 lines of code (using the StarCore SC140 as a recent example).

This highlights the shift that has occurred in terms of software distribution (from centralised to distributed software).

Consider that an AXE switch has **20 million lines of code**.

In a 3G cellular network, one thousand subscribers will have (between them) **one trillion lines of code** (code bandwidth), 10 terabytes of distributed storage (memory bandwidth) and one billion MIPS (processor bandwidth).

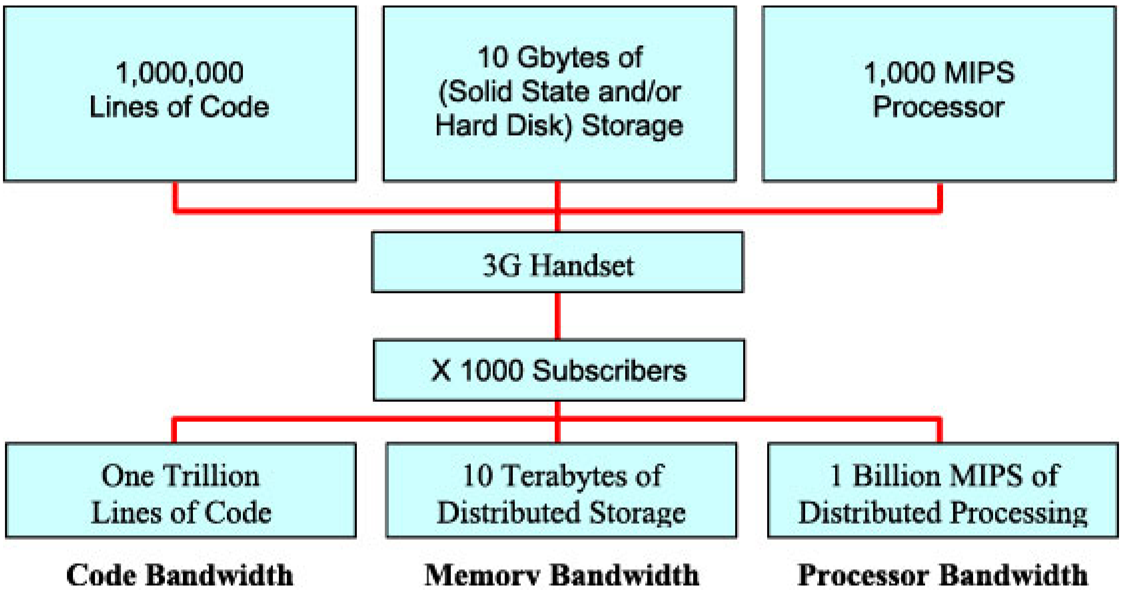

Figure 3 - Distributed Code, Memory

The addition of these **software components** (into the handset) changes what a user can do and what a user expects from the device and from the network to which the device is connected.

We can use an analogy from the computer industry to show how software influences user behaviour.

We know that the software in a PC influences the way in which we use the PC. A present example would be the rapid growth in instant messaging (peer to peer file exchange) which in turn has an impact on network loading and routing **and** network performance expectations (the interactive exchange of complex data files).

The composition of the code in a 3G handset will determine how a 3G network is used, ie software form factor and functionality will determine application form factor and functionality, which in turn will determine network form factor and functionality.

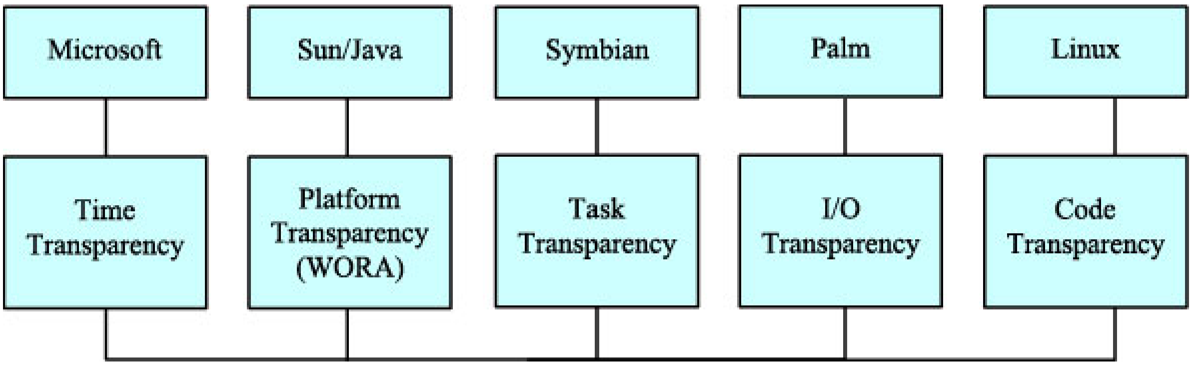

Figure 4 - Handset Software Form Factor and Functionality

Figure 4 shows five possible application layer software options for a 3G cellular handset.

Each of the options has a number of unique properties.

Microsoft have traditionally majored on 'time transparency' - backwards compatibility with earlier (Microsoft) software.

Sun and Java have traditionally offered platform transparency - the write once read anywhere message.

Symbian have offered task transparency - the ability to support multi-tasking. Palm have offered input/output transparency (eg handwriting recognition, bar code inputs) and Linux have offered code transparency (open source software).

Early architectural decisions can sometimes affect future functionality - Palm's use of an event loop to handle task management produced a memory and processor efficient platform but made multi-tasking tricky.

The Symbian, Microsoft and Java platforms promote their extensive use of multi-threading as a basis for being able to support multiple tasks in parallel.

Multiple tasking includes the ability to support parallel voice, image, video and application encoding in a complex data file.

The dynamic range of the voice, image and video streams depends on the encoding used. For example the voice encoder rate could be any one of 8 source rates between 4.75 to 12.2 kbits, the J-PEG encoder rate would depend on the image bandwidth to be sent, the M-PEG encoder rate would be determined by the implementation (for example it could be constant rate variable quality or variable rate constant quality).

Either way, the summation of the individual channel streams determines the dynamic range (information rate excursion boundaries) of the composite bandwidth which then has to be modulated on to the (radio) channel.

The dynamic range can be reduced by buffering, effectively smoothing the bursty data rate prior to transmission, but this will introduce delay.

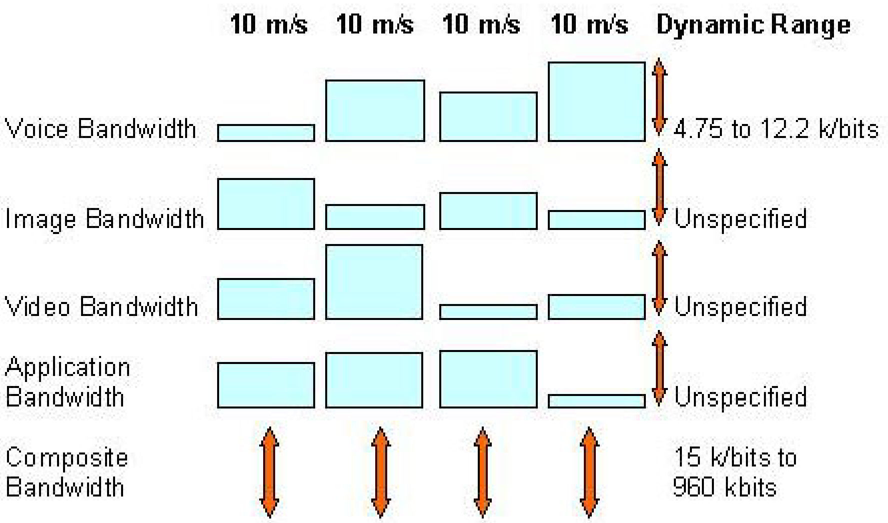

Figure 5 - Dynamic Range of a Composite Data Stream from a Single User

Note also that there will likely be time interdependency between multiple channel streams that will need to be preserved across the radio interface and into and through the core network.

The assumption is that each separate channel stream is separately encoded on to the radio physical layer - ie voice is on one code channel, image on another, video on another, application and data streams on another. The 3GPP specification (3GPP1) specifies that up to 6 channel streams per user are available on the uplink (if supported by the handset) and up to 6 channel streams per user on the downlink (if supported by the base station) - a channel stream can be anything from 15 kbits to 960 kbits (although higher bit rates may be multiplexed across multiple code channels).

The objective across the radio physical layer is to preserve any time independency existing across the multiple channel streams.

Time interdependency then has to be preserved as the traffic moves into the core network.

If the core network is a packet network, each traffic stream needs to be treated identically in terms of routing and priority or needs to be time stamped and re-clocked. Re-clocking requires buffer bandwidth to be made available and introduces delay and delay variability to the composite channel stream, ie some of the properties (and value) of the original data stream will have been lost.

The solution is to circuit switch or cell switch the data or to do virtual circuit switching on a packet network. Either way, the data will cost more to deliver.

SUMMARY

Handset software form factor and functionality determines application form factor and functionality which in turn determines network form factor and functionality.

It is the job of the physical layer (radio bandwidth) and network and transport layers (network bandwidth) to preserve the 'properties' of the original input data streams. When multiple data streams are encoded in parallel, the 'properties' of the composite data may include time inter-dependence that needs to be preserved to the point of receipt.

Buffering can be used to reduce the burstiness of the data being sent and/or to re-establish time inter-dependencies by re-clocking multiple data streams after decorrelation (ie in the receiver). However this introduces delay and delay variability (which may or may not be important).

Software components add value in terms of application complexity - software allows us to perform complex tasks including complex **multiple** tasks in parallel. This in term places particular demands on the networks to which these (complex) devices are connected.

In our next Hot Topic, we will explore the process of 'network discovery' and 'network negotiation' (declarative content and declarative applications), how declarative applications can be used to determine network performance requirements.

About RTT Technology Topics
RTT Technology Topics reflect areas of research that we are presently working on.

We aim to introduce new terminology and new ideas to clarify present and future technology and business issues.

Do pass these Technology Topics on to your colleagues, encourage them to join our Push List and respond with comments.

Contact RTT
RTT, the Shosteck Group and The Mobile World are presently working on a number of research and forecasting projects in the cellular, two way radio, satellite and broadcasting industry.

If you would like more information on this work then please contact
geoff@rttonline.com
00 44 208 744 3163