

Seattle Overview: iAnywhere® Blue SDK 4.0 Description

TABLE OF CONTENTS

- 1 Introduction
- 2 System Operation
- 4 Integration into a System
- 5 Application Development
- 5 Phased Rollout

INTRODUCTION

Bluetooth technology continues to evolve with the planned release of the Bluetooth Core Specification 3.0 + HS (High Speed) in Q2 of 2009. With the inclusion of a secondary Controller running an 802.11 MAC/PHY (and/or Ultra Wideband MAC/PHY in the future), data rates can be increased by 15 times that of a standard Bluetooth link .

Increased bandwidth enables use cases such as the synchronization of playlists between PC and MP3 players, photo downloads between PCs and cameras, the transmission of video and the use of PAN (Personal Area Networking) that can support peer to peer communication for applications involving toll road payments or even traffic information for navigation systems. All other Bluetooth profiles can take advantage of this increased bandwidth as well if required.

Interest in Seattle, as this release is referred to, is now beginning to materialize in diverse markets such as PCs, cell phones, consumer electronics, and in-car telematics. Demand for creating solutions, however, is originating with the integrated circuit (IC) manufacturers that have plans in supporting both Bluetooth and 802.11 radio Controllers either in the form of separate IC devices or as combined silicon. In support of this effort, the iAnywhere Blue SDK v 4.0 will be making minimal changes to the existing application programming interface (API) structures already in place. Differences that will be introduced will most likely be found in the following areas:

- 1) Existing profiles will only require minimal change to take full advantage of the high speed capability.
- 2) Connection Management will undergo modification to accommodate the inclusion of an Alternate MAC/PHY (AMP) Manager. The AMP Manager facilitates connections made over high speed AMPs yet for the most part, the AMP Manager remains hidden unless applications require access. Currently, the BlueSDK provides a Management Entity module that supports the establishment of connections through "lower level" APIs. The continued use of the ME will be discouraged in favor of using the Connection Manager which acts not only to consolidate ME calls to simplify generic access application development but will also offer some intelligence in managing connections.
- 3) A software interface encapsulating access to AMP Controllers will be added to the BlueSDK 4.0 offering. Sybase iAnywhere has announced that we are currently working with a hardware partner, Marvell, to integrate support for their 802.11 AMP Controller which is planned for early release.
- 4) Integration into the operating system is expected to remain similar to what is currently in place.

Upgrading to Seattle from the current BlueSDK v3.x is being developed in such a way as to require minimal changes to existing application software, making it an attractive option for customers wishing to upgrade. Though existing applications will operate over AMP Controllers, they may need modification to fully achieve the performance that is possible via a high speed AMP.

SYSTEM OPERATION

A typical Bluetooth architecture is shown in Figure 1 whereby data passes through the L2CAP (logical link and adaptation protocol) and out through a Controller (Basic Rate (BR)/Enhanced Data Rate (EDR) and/or AMP) to the remote device. L2CAP can create multiple channels each defined with a channel identifier (CID). L2CAP supports both connection oriented or connectionless links, though connectionless can currently operate over only the BR/EDR Controller. L2CAP supports both dynamic and fixed CIDs

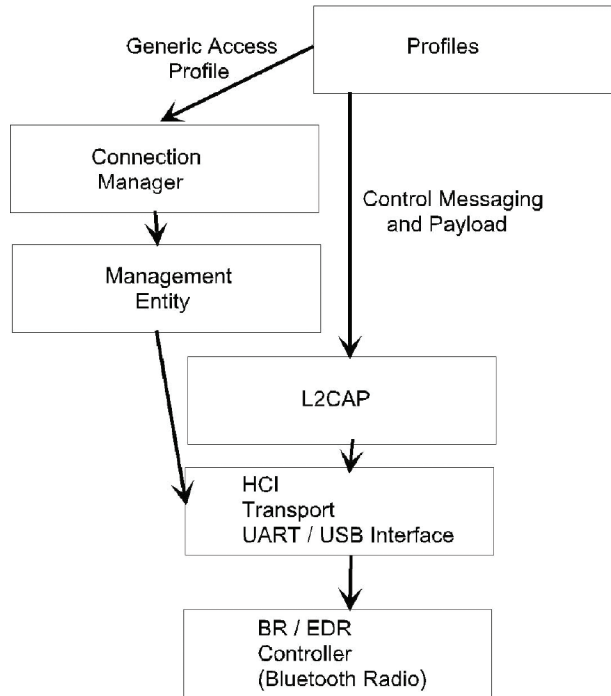


Figure 1: Bluetooth Architecture

Figure 2 shows the Seattle architecture with the addition of several new modules. The AMP manager controls the establishment of connections through either BR/EDR and AMP Controllers and route payload data through the appropriate physical layers. The initial connection between devices is made using the BR/EDR Controller, in much the same way that all Bluetooth connections are made. Essentially, no change has been introduced to this procedure. With such a connection in place, the AMP Manager is then invoked to determine the types of AMPs that are supported on the remote side of the link. The AMP Manager protocol uses an L2CAP channel with a fixed CID running over the BR/EDR Controller.

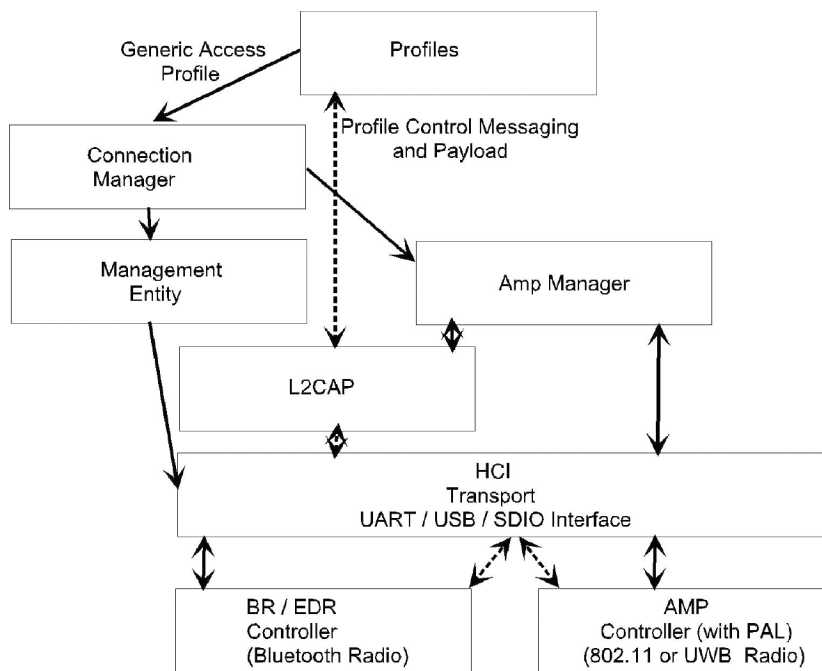


Figure 2: Seattle Architecture

Assuming the remote device supports an AMP that is compatible with the local device, the AMP manager can then establish an AMP physical link to the remote device by issuing AMP manager commands to the remote device and control commands to its local AMP.

Once the physical link between AMP Controllers is established it can be used to support peer to peer profile communications. Since there are multiple physical links between the devices (BR/EDR and AMP) the data can be routed through either a physical link or by moving the L2CAP channel from one Controller to the other. In the case where data channeled through the AMP and the AMP physical layer is disrupted, the data can be re-routed back through the BR/EDR Controller to maintain connectivity; this is handled in L2CAP through a “move channel” operation.

The current Bluetooth stack architecture will be modified to support both AMP Management functions and the need to route data to the correct Controller residing beneath the stack. This requires changes in the L2CAP layer. Additional changes to this layer will add 2 more modes of operation: Enhanced Retransmission mode (an enhanced version of the existing Retranmission mode) which fixes a number of problems with Retranmission mode and Streaming mode (for isochronous traffic that implements flush timeouts on the transmit side and packet flushing on the receive side if the sender transmits faster than the receiver can handle). These modes are to be used if both sides of the link support the capability.

Quality of service is also improved with the addition of the extended flow specification which allows applications to define Best Effort and Guaranteed channels (guaranteed latency, guaranteed bandwidth or both) that can run over either the BR/EDR Controller or AMP Controllers. Extended flow specification feature allows devices to negotiate a data rate for a particular channel which is based on packet size and frequency of packet arrival.

Less apparent are the changes that will be made to modify the existing memory management structures to accommodate data routing and the efficient handling of high speed data passing over the AMP Controller.

Figure 3 provides a view of the BlueSDK 4.0 stack, identifying modules that are either being added or updated and those that should not be changed at all. Currently, the expectation is that upper layer profiles will remain the same, allowing legacy applications currently integrated with BlueSDK 3.0 (Lisbon: Bluetooth core specification v2.1 + EDR) to be easily ported to the new Seattle structure. The hardware interface to the BR/EDR Controller remains intact as well, although there may be changes to accommodate a more efficient memory scheme.

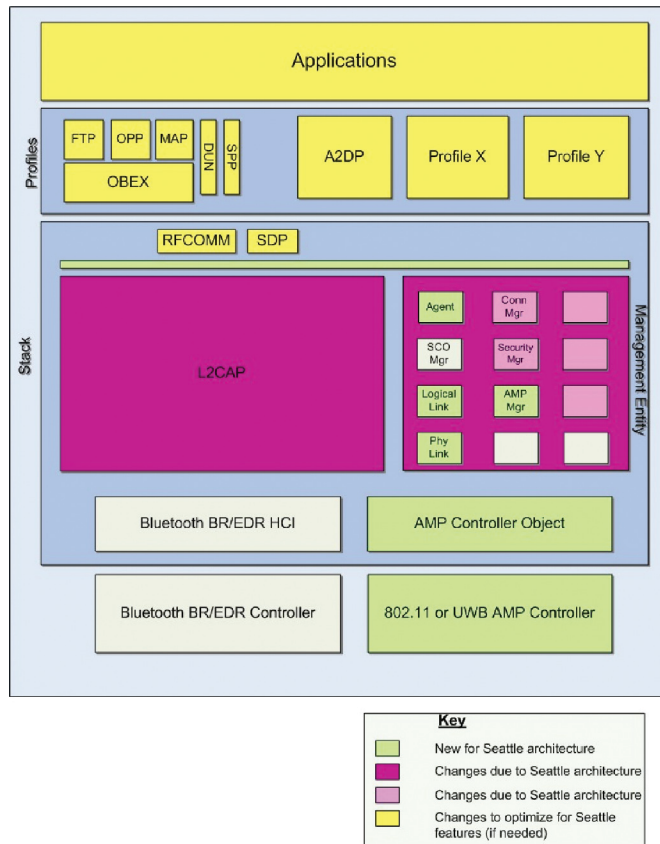


Figure 3: iAnywhere BlueSDK 4.0 Architecture

New modules include AMP Controller objects that are responsible for managing AMP Controllers. Also, the AMP Manager is added and is responsible for the discovery of other AMP capable devices and establishing physical links over AMP Controllers. Logical and physical link classes are added to encapsulate the the links that can exist between AMP Controllers. The BR/EDR Controller has similar concepts though there is only one logical link per physical link for a BR/EDR Controller while there may be multiple logical links per physical links for an AMP Controller.

Security is updated to handle security needs for both BR/EDR and AMP Controllers. An Agent class is required and encapsulates the creation of an AMP physical link and manages how L2CAP channels are created and moved between different Controllers. Finally, there is a connection manager that offers higher level APIs to the developer, thereby simplifying connection management for the BR/EDR Controller.

At this time, APIs associated with the new and modified modules are not defined to the point where development can be supported. Prototyping of the modules is underway for verification at the June Unplugfest to prove interoperability.

INTEGRATION INTO A SYSTEM

Integrating the protocol stack involves developing additional software in 5 unique areas:

- Operating system (OS) interface
- BR/EDR Controller
- AMP Controller
- Application (including connection management)
- Dataset interface (for security)

The stack has been designed to integrate into any OS easily. Currently there are several APIs that have to be populated with OS specific code, each being as simple as one line without exception handling. A stack task must be written as well, which takes on a simple structure whereby the protocol stack is called periodically enabling it to process requests and data.

Interfacing to BR/EDR and AMP Controllers requires the development of drivers. BR/EDR drivers have been developed and reside beneath the transport layer; they support both USB and UART type devices. Modification of drivers that will be provided in the SDK is considered an easy task.

Developing an interface to an AMP Controller can take several flavors. At this time, an SDIO interface is used to communicate with an HCI based AMP Controller (provided by Marvell). This requires specific hardware and an SDIO driver that will be developed.

Other AMP Controllers may have a part or all of their PAL operating in the host. This requires:

- 1) The development of code to communicate directly to the PAL software,
- 2) Porting of the PAL software, and
- 3) A driver to talk to the AMP Controller using its proprietary interface.

The connection manager will be overhauled, providing the developer with a set of high level APIs that control the BR/EDR connections made to remote devices.

APPLICATION DEVELOPMENT

As has been discussed, the majority of changes to the protocol stack affect the internal protocols and connection management schemes. To make use of the high speed AMP, additional (or modified connection management) APIs will have to be accessed. Also, adaptation to existing profiles will be required to make use of the high speed AMP as well (e.g. A/V Profiles, PAN profiles). In many cases however, existing profiles will not need to take advantage of the HS capability and will remain unchanged. Backward compatibility will be maintained for such profiles (e.g. HF, SPP, PBAP, etc.).

It is expected that developers will have to make changes to the connection management application given that APIs will be modified in this module. Sybase iAnywhere strongly discourages the continued use of the Management Entity to create connections with devices.

PHASED ROLLOUT

Development of the Seattle stack has been underway and will be released in phases during 2009, following the adoption of the Bluetooth 3.0 core specifications. The generic AMP and 802.11 PAL are expected to be accepted in Q2 2009. Sybase iAnywhere will be attending interoperability events as they are held to prove that our implementation of the specifications will work well with other vendors' implementations. The phased roll-out of our Seattle release is planned as follows:

- March IOP – Basic/simplified AMP functionality
- Unplugfest 33 June 8-12, 2009 – Working Profile demo/prototype (file transfer)
- July 15, 2009 – Q2 Pre-Release #1 (limited functionality, API's not frozen)
- September 2009 – Q3 Pre-Release #2 (API's stable)
- Unplugfest 34 October 5-9, 2009 – Demonstrate interoperability of full stack
- November 2009 – Q4 Official Release of Seattle Stack



CONCLUSION

Seattle development is a significant undertaking that introduces additional radio hardware to increase data bandwidth when required. Initially an 802.11 MAC/PHY is being integrated and verified. Changes to applications will be kept to a minimum, yet are expected in generic access (connection management), A/V (audio/visual) and PAN profiles. Other profiles are expected to remain intact, yet can always be modified to use the high speed AMP option.

The operating system interface is expected to remain the same. Hardware interface code will have to be slightly modified to accommodate the hardware design implementation on the embedded system being built.

In conclusion, Sybase iAnywhere is investing heavily in this new technology, first in its efforts in developing the core L2CAP specification and secondly, in implementing this and all other specifications associated with Bluetooth Core Specification v3.0. To give customers a head start, a phased release of Seattle is planned, allowing developers an early chance to evaluate a demo version of this software and then build upon this with additional software releases providing greater functionality and stability.