

Development of a Service Discovery Architecture for the Bluetooth Radio System

Christian Schwingenschlögl, Anton Heigl

Technische Universität München (TUM), Institute of Communication Networks

Arcisstr. 21, D-80333 Munich, Germany, e-mail: schwinge@lkn.ei.tum.de

Arcisstr. 21, D-80333 Munich, Germany, e-mail: toni.heigl@bmw.de

Abstract

Bluetooth 1.0 was published in 1999 as an industry standard for short-range wireless data and voice communication. Application profiles cover cordless telephony, wireless access to printers, fax machines or LANs, Personal Area Networking and more. In order to handle this variety of services and nevertheless guarantee interoperability and auto-configuration of different devices from different manufacturers, the Bluetooth specification contains the so-called Service Discovery Protocol (SDP).

This paper discusses the characteristics of SDP, its requirements and its limitations. Furthermore, it describes the implementation of a system architecture for the complete Service Discovery Application which includes client and server, a user interface, the SDP protocol itself and a control module for the Bluetooth links.

1 INTRODUCTION TO BLUETOOTH

The Bluetooth technology [1] is an open specification for wireless communication of data and voice. Devices that will be equipped with this technology can establish short-range radio links to each other for multiple purposes. Bluetooth is based on a frequency hop scheme with a hop rate of 1600 per second. It uses the unlicensed ISM band between 2.402 GHz and 2.480 GHz. The gross data rate is 1 Mb/s and can be used for synchronous voice channels and asynchronous data channels at the same time.

Every Bluetooth device can initiate connections. Initially, a device has to collect information (e. g. the unique 48 bit Bluetooth addresses) from other Bluetooth modules in range. This is done during a so-called "Inquiry Procedure". After that, a device can

connect with up to seven other devices by performing a "Page Procedure". The connected devices belong to a "Piconet", while the initiating device becomes master of this piconet. The master determines the frequency hop sequence of the piconet and polls its slaves before they are allowed to respond.

It is important to emphasize that initially all Bluetooth devices have the same status. Any device can contact any other and become a master this way. A device which is responding to a connection request automatically becomes slave. A Bluetooth network does not need any central administration, but relies on a self-organizing architecture with distributed intelligence. Therefore it can be considered as an implementation of what is called a Mobile Ad Hoc Network. [2]

One device can be a member of several piconets at the same time, e. g. as a master in one piconet and as a slave in another. This concept is called "Scatternet" within the Bluetooth specification.

Bluetooth is designed as a low-cost product for the mass market. Mobile phones, headsets and notebook computers will be among the first products to be equipped with Bluetooth technology. But applications are not limited to simple cable replacement, even if this was the initial idea behind Bluetooth. The current specification offers a much wider functionality and also gives way to further technological extensions.

2 PROTOCOL STACK

2.1 Core Protocols

Bluetooth not only consists of a physical link specification, it actually includes a complete software framework and some compatibility recommendations

as well. An overview of the Bluetooth protocol architecture is depicted in figure 1. All layers from Bluetooth Radio up to RFCOMM and furthermore SDP are part of the Bluetooth specification. For the other protocols (e. g. WAP), the Bluetooth specification gives instructions in order to guarantee interoperability.

The following paragraphs describe the basic tasks of the most important Bluetooth protocol layers.

2.1.1 Baseband, Link Manager Protocol (LMP) and Host Controller Interface (HCI)

These parts of the Bluetooth Specification are generally integrated in the Bluetooth module itself. All other protocols have to run on the host, i. e. on the platform that is controlling the Bluetooth module.

The Baseband protocol enables the physical RF link to form a piconet. It provides ACL and SCO links between the different Bluetooth units.

The Link Manager Protocol (LMP) is responsible for link set-up and security aspects like authentication and encryption. It also controls the power modes and the connection states of a Bluetooth unit in a piconet.

The Host Controller Interface (HCI) provides standardized access to the Bluetooth module. The higher layers on the host can make use of the HCI commands to create connections, for example. The signals from the Bluetooth module to the host are called HCI events.

2.1.2 Logical Link Control and Adaptation Protocol (L2CAP)

Once the connection is established, L2CAP provides data services to the upper layer protocols with multiplexing capability, segmentation and reassembly operation. L2CAP can handle data packets up to 64 kB in length and supports ACL links only.

2.1.3 RFCOMM

RFCOMM is a serial line emulation protocol. It was adopted from the ETSI TS 07.10 specification [3]. This protocol emulates RS232 control and data signals, providing transport capabilities for upper level services. A variety of non-Bluetooth protocols can use RFCOMM as transport mechanism, e. g.

TCP/IP over PPP or the object exchange protocol OBEX.

2.2 Application Profiles

The Bluetooth specification currently includes 14 application profiles that define the requirements for certain use cases, e. g. the Fax Profile or the LAN Access Profile. Beyond these application profiles, there are generic profiles that serve as a base for other applications, e. g. the Generic Access Profile or the Serial Port Profile.

The profiles represent an important part of the interoperability requirements, as all manufacturers of devices for a certain use case are obliged to fulfill the respective profile. The Bluetooth Special Interest Group is still working on additional profiles for further use cases.

The use of the service discovery mechanisms is also regulated in a specific profile, the Service Discovery Application Profile which was an important prerequisite for the implementation in section 3.2.

3 SERVICE DISCOVERY ARCHITECTURE

3.1 Service Discovery Protocol (SDP)

As computing continues to move to a network-centric model, finding and making use of services that may be available in the network becomes increasingly important. This problem – commonly known as Service Discovery – is widely recognized; many companies, standards bodies and consortia are addressing it at various levels in various ways. Bluetooth Service Discovery Protocol (SDP) addresses service discovery specifically for the Bluetooth environment. It is optimized for the highly dynamic nature of Bluetooth communications. SDP focuses primarily on discovering services available from or through Bluetooth devices. SDP does not define methods for accessing services. Once services are discovered with SDP, they can be accessed in various ways, depending upon the service. This might include the use of other service discovery and access mechanisms such as those mentioned above. SDP provides a means for other protocols to be used along with SDP in those environments where this can be beneficial. While SDP can coexist with other service discovery protocols, it does not require them. In Bluetooth environments, services can

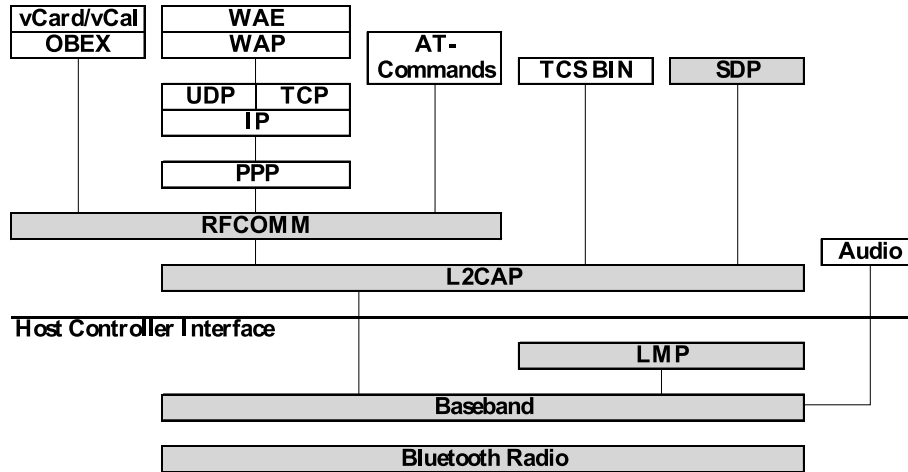


Figure 1: Bluetooth Protocol Stack

be discovered using SDP and can be accessed using other protocols defined by Bluetooth.

The Service Discovery Protocol (SDP) provides a possibility to query the network for certain services, characteristics of services or other service information. The service discovery application profile defines the protocols and procedures that shall be used by a service discovery application on a device to locate services in other Bluetooth-enabled devices using the Bluetooth Service Discovery Protocol (SDP).

The service discovery application described by that profile is a specific user-initiated application that provides the following service inquiries:

- Search for services by service class
- Search for services by service attributes and
- Service browsing.

The services in SDP are classified in a hierarchical structure of service classes. A service is an instance of a service class. The specification includes basic service classes, but implementations are not limited to these classes. Each service is represented by a service record, which is in turn a list of service attributes (see figure 2). Each attribute has an ID and a value of a specific data type. This list has two mandatory entries: the ServiceRecordHandle, which is a unique ID within the SDP server(=the Bluetooth device offering services), and the ServiceClassIDList, which contains the path of service classes, i. e. the service class that the specific service belongs to and all superior service classes. All other entries are optional. They can provide information about the instance of service (ServiceID), the required protocols for utiliz-

| Service Record | |
|------------------------|-----------------|
| attribute ID | attribute value |
| ServiceRecordHandle | <int> |
| ServiceClassIDList | <UUID(s)> |
| ServiceID | <UUID> |
| ProtocolDescriptorList | <sequence> |
| BrowseGroupList | <UUID(s)> |
| DocumentationURL | <URL> |
| ... | |
| | |
| | |

Figure 2: Data Representation in the Service Record

ing the service (ProtocolDescriptorList), a link to a documentation file (DocumentationURL) and so on.

The attribute values can have different data types. Most entries have the type UUID (universally unique identifier) or a set of UUIDs. This is a 128-bit unsigned integer value generated in a particular way [6]. The UUIDs also play an important role in the search transactions of SDP as described below.

The Service Discovery Protocol conveys information from the SDP server(the Bluetooth unit providing a service) to the SDP client(the Bluetooth unit utilizing a service). Generally, any Bluetooth device can act as server and client, even at the same time. The architecture of this protocol is based on a simple request-response scheme. There are three different cases, how SDP may be used:

1. ServiceSearch Transaction.

This transaction searches for services. The request includes a search pattern consisting of

UUIDs; the response is a list of ServiceRecordHandles.

2. ServiceAttribute Transaction.

If the client has obtained a ServiceRecordHandle from the server, it can request information about further service attributes. The ServiceRecordHandle is sent in the request and a list of service attributes will be in the response message.

3. ServiceSearchAttribute Transaction.

This transaction combines the capabilities of the others in one single request.

The goal of the development was a service discovery application according to the Bluetooth profile. The developed software comprises the service discovery application itself, the Bluetooth SDP layer for the server and the client, and a module for controlling the baseband and link manager function via the Host Controller Interface (HCI).

3.2 Implementation

The Institute of Communication Networks together with the BMW research department formed a working group in order to explore the field of Ad Hoc Networking. Among other technologies, one focus was set on Bluetooth. Within the thesis [4], an implementation of a Bluetooth service discovery architecture was developed using SDL (Specification and Description Language) [5], a formal language standardized by the ITU. In figure 3 an overview of the general architecture of the Service Discovery Client System can be seen. The three main blocks are:

- the service discovery application block which handles user input and output and stores retrieved service information,
- the SDP block which implements the actual SDP layer, i. e. it transmits and receives SDP messages over the L2CAP layer, and
- the BT module control block which controls the Bluetooth module using HCI commands in order to set up and tear down physical links prior to the SDP request.

The message sequence chart in figure 4 shows a complete service discovery process. The process of service discovery consists of four different stages:

1. During an inquiry procedure, the device that is looking for services (in the following called the

SDP client), gets information about other devices in the vicinity, i. e. about possible SDP servers. The following items are repeated for any of the discovered devices. A selection can be made based on the parameter 'class of device', that is received during the inquiry procedure.

2. The SDP client establishes a connection on LMP level using HCI commands. This step can be skipped, if there is already a connection existing.
3. The SDP client establishes an L2CAP connection dedicated to the use by the SDP layer.
4. The actual SDP requests can be passed through the L2CAP channel.

The disconnect processes on the L2CAP and LMP layer are not shown in the message sequence chart.

The Service Discovery Server System (fig. 5) is designed to generate the appropriate responses to the requests from the client system based on the information stored in the server database. It consists of the actual SDP(Server) block and the Database block.

The database can be modified by specific user commands. It contains all the service records (see figure 2) and the applicable service attributes for each service.

The SDP(Server) block handles incoming L2CAP connections, reads and decodes the SDP requests and generates response messages based on the database information.

3.3 Conclusion

The development of the Service Discovery Architecture had to fulfill the following requirements:

- Compliance with the Bluetooth SDP specification and the Bluetooth Service Discovery Application Profile [1].
- Usage of pre-defined interfaces to HCI and L2CAP.
- Representation of the SDP data concept.
- Definition of an appropriate user interface.

Despite these requirements there are various possibilities for variations and extensions of the system. The Bluetooth specification allows, for example, the definition of further service classes and service attributes by any user, which is an important factor in the concept of distributed self-organization and auto-configuration. The developed implementation

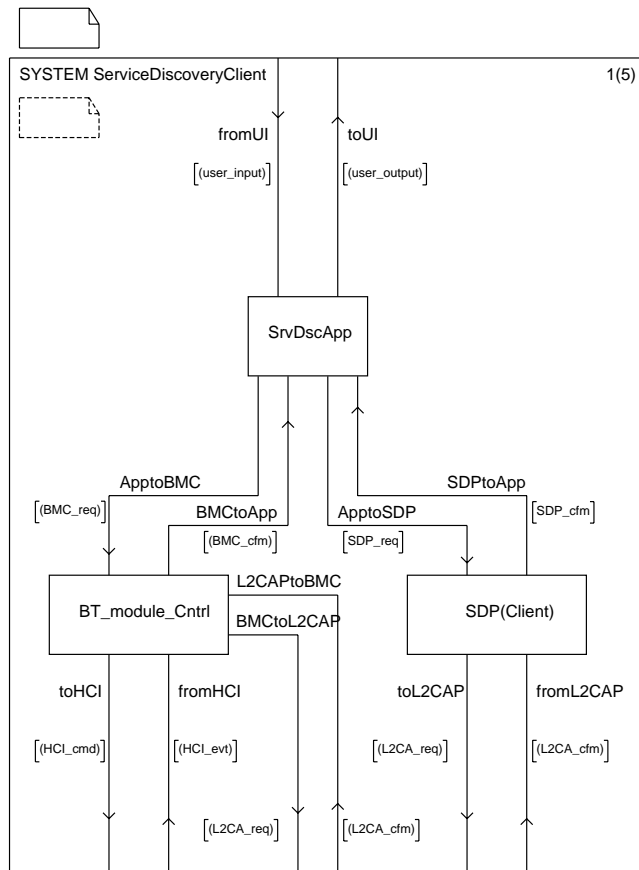


Figure 3: Architecture of the Service Discovery Client according to the Bluetooth Service Discovery Profile

was also designed in order to support modifications, e. g. multiple parallel SDP requests or further user commands.

Concluding, it can be said that Bluetooth in general and SDP in particular are an important contribution to the ongoing research work in the field of Ad Hoc Networking.

4 APPLICATION SCENARIO

The user benefit of Service Discovery within the Bluetooth Radio System can be demonstrated by various applications. Only one scenario shall be mentioned here as an example in order to make the user experience more obvious.

In few years, different electronic devices in a conference room are likely to be equipped with Bluetooth. Say, this is the case for the video beamer, a LAN access point and a printer. The participants of the

conference bring their PDAs and notebook computers into the room which also have an integrated Bluetooth module.

The Service Discovery Application on the notebook computer of the first speaker is able to display all available services in the surroundings. As the user is trying to connect to the video beamer, he can also search for members of the service class "video beamer". Possibly, other devices in proximate rooms are also responding. It is obvious that a sensible use of other service attributes, e. g. defining the location, is required.

Another solution may be to implement a Service Discovery Application on the video beamer. The conference master can use this application in order to discover the computers of the conference members and give only them the access rights to display their presentations.

The Service Discovery Application can be a very powerful tool, if designed in an appropriate manner.

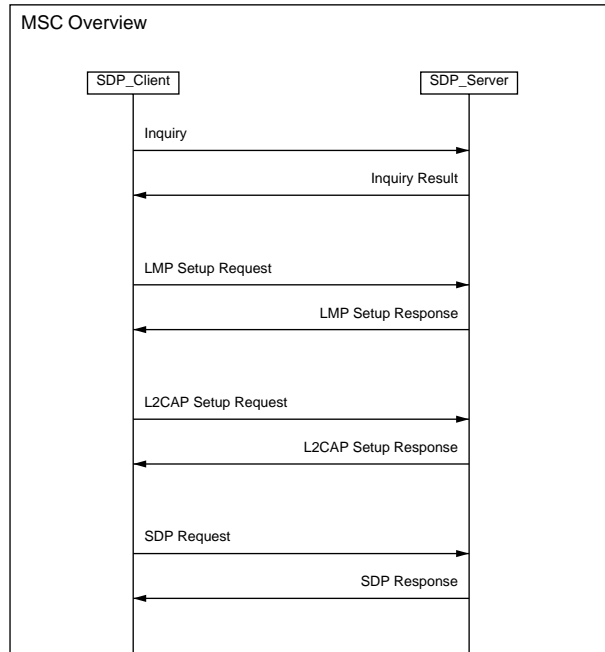


Figure 4: Overview of the different transaction within a service discovery process

Therefore, the underlying architecture has to fulfill universal requirements. Bluetooth SDP and the developed system should be able to serve as a good reference further extensions that are more application-oriented.

References

[1] Bluetooth Special Interest Group: Bluetooth Specification Version 1.0B, <http://www.bluetooth.com>, 1999.

[2] Internet Engineering Task Force (IETF): Mobile Ad Hoc Networks Charter, <http://www.ietf.org/html.charters/manet-charter.html>, 1999.

[3] European Telecommunications Standards Institute (ETSI): Technical Specification TS 101 369 V6.3.0 (1999-03), <http://www.etsi.org>, 1999.

[4] Heigl, Anton: Installation of a Bluetooth Test Environment and Development of a Service Discovery Architecture, Diploma Thesis, Institute of Communication Networks, Technische Universität München, 2000.

[5] International Telecommunication Union: ITU-T Recommendation Z.100, Specification and De-

scription Language (SDL), <http://www.itu.int>, 1994.

[6] International Organization for Standardization/International Electrotechnical Commission: ISO/IEC 11578 standard, Information technology – Open Systems Interconnection – Remote Procedure Call (RPC), <http://www.iso.ch>, <http://www.iec.ch>

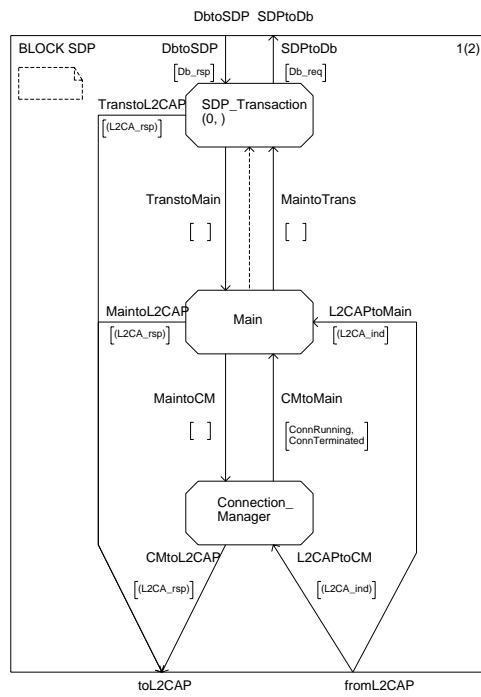


Figure 5: Architecture of the Service Discovery Server System