Accessing CORBA Services from Bluetooth Mobile Terminals

Sami Merovuo, Ari Valtaoja, and Andrei Kotchanov
Lappeenranta University of Technology, P.O .Box 20, 53851 Lappeenranta, Finland

{sami.merovuo, ari.valtaoja, andrei.kotchanov}@lut.fi

Abstract The recent development of mobile communications and mobile devices has created a demand for running distributed applications on mobile devices. CORBA is currently the most widely used middleware solution in fixed networks, providing an open, widely applicable and platform-independent technology that hides most of the complexity in distributed systems. However, CORBA is designed for fixed networks and is not suited for networks containing wireless links. The standardization body of CORBA has recently released a new standard for wireless access and terminal mobility in CORBA which provides a framework for the use of CORBA on wireless networks, and the mechanisms to hide mobility. However, it does not deal with what kind of wireless technologies and networks the suggested framework can be used for and what kind of requirements it sets for them. In this paper, we discuss these requirements, and propose ways in which the framework can be bridged together with a real wireless network technology, Bluetooth. This paper also presents the problems of Bluetooth from the viewpoint of wireless CORBA and suggests some possible solutions for them. This paper proposes, as a result, the architecture for the use of wireless CORBA over Bluetooth.

Keywords: Bluetooth, CORBA, Wireless networks

1. INTRODUCTION

The recent development of mobile communications and mobile devices has created a demand for running distributed applications on mobile devices. In the future, users will even expect to be able to use the same services in both, stationary and mobile devices. Common Object Request Broker

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35584-9_19

O. Martikainen et al. (eds.), *Smart Networks*

Architecture (CORBA) is currently the most widely used middleware solution in fixed networks, providing an open, widely applicable and platform-independent technology that hides most of the complexity in distributed systems.

However, CORBA, as it exists today, is not perfectly suited for wireless networks because certain assumptions, which are not valid for networks containing wireless links, have been built into CORBA. The standardization body of CORBA, the Object Management Group (OMG), has recently released a standard, the *Wireless Access and Terminal Mobility in CORBA specification* [1], which provides a framework for the use of CORBA on wireless networks, and the mechanisms to hide mobility.

A new wireless technology, Bluetooth, has drawn attention in recent years and it is expected to be the most widely used short-range communication technology between mobile devices in the future. First Bluetooth cards have already entered the market, and the real breakthrough is expected to happen in the near future. The high probability that Bluetooth will be widely used technology, especially on access networks, means that there is a need to provide CORRA services through Bluetooth connections. More than that, making wireless CORBA and Bluetooth work together will also speed up the breakthrough of both technologies.

2. WIRELESS CORBA REQUIREMENTS FOR NETWORK TECHNOLOGY

The main component of CORBA is the Object Request Broker (ORB) which enables communication between client and server objects. Communication in CORBA is transparent, allowing a client to make an object request from a server without having to know where the server is in a distributed network. In a wireless world mobile terminals can access CORBA services from different locations, and perform connection establishment and disconnection at any time. This makes transparent communication more difficult to implement. The wireless CORBA specification provides a framework which enables this transparency by hiding the mobility of the mobile ORBs. The framework does not require any modifications to be made to non-mobile ORBs which means that ordinary ORBs on fixed network do not have to implement the wireless CORBA specification. However, the specification does not discuss on what kind of wireless technologies and networks the suggested framework can be used and what kind of requirements it sets. In order to bring wireless CORBA to some particular wireless network technology, these requirements

have to be defined, and fulfilled by the used technology. These requirements are listed below:

- the existence of mechanisms to detect transport end-points on the link, the network and the transport layers [1].
- the ability of a data terminal to connect to, and disconnect from the transport end-point in a reasonable time.
- the ability to detect connection loss in a reasonable time.

In addition, wireless CORBA contains some optional features that demand more from the underlying technology. The most important of these features is the handoff process, which allows a mobile device to move from one access point to another without losing the CORBA connection. In order to implement the handoff process, there should exist mechanisms in the network technology that inform the data terminal and the access point when the handoff process should be started. The indication can be obtained, for example, from the user, in the case of manual handoff, or from the network, in the case of automatic handoff. To fully implement handoff, the network technology should also allow the establishment of two simultaneous connections between the data terminal and the access point (see chapter 4.2).

3. WIRELESS CORBA CONNECTION

The General Inter-ORB Protocol (GIOP) is an element in CORBA which specifies standard transfer syntax and a set of message formats for communications between ORBs. The GIOP is just an abstract protocol and cannot, therefore, be used to communicate directly between ORBs, although a concrete implementation of GIOP in specialized mapping for some real transport protocol is required. In ordinary CORBA, there exists an end-to-end TCP/IP connection between the client and the server, and the GIOP messages are transmitted using the Internet Inter-ORB protocol (IIOP). In the Wireless CORBA approach, the GIOP connection between the client and server consists of two different connections: one between the mobile host and the access point (Access Bridge), and another between the access point and the other end of the CORBA connection that can be a host in a fixed network or a mobile host (see Figure 1). In order to maintain the interoperability between ordinary ORBs and mobile ORBs, the GIOP connection in the fixed network should be a normal IIOP connection over TCP/IP. Another connection over a one-hop wireless link (e.g. Bluetooth) does not necessarily have to be over TCP/IP which is not a very convenient solution for wireless links without any extensions [2]. Rather, a more optimized wireless link-specific protocol tuned for better performance can be used.

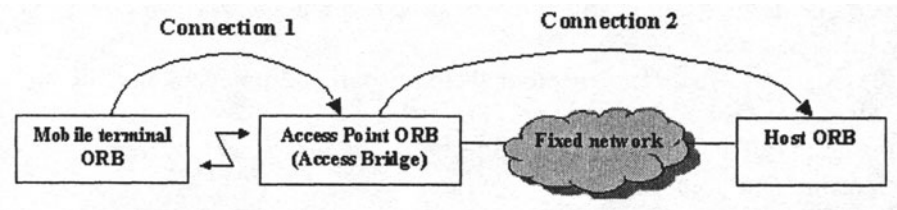

Figure 1. GIOP connections in wireless CORBA

Wireless CORBA does not provide mapping of GIOP messages to some concrete protocol like IIOP in ordinary CORBA. Instead, it specifies a tunneling protocol which encapsulates and decapsulates GIOP messages over the real transport protocol on the wireless media. The GIOP Tunneling Protocol (GTP) is an abstract, transport-independent protocol which defines the message formats for establishing, releasing, and re-establishing the tunnel, but also for transmitting GIOP messages. It also defines the messages for maintaining the GIOP connection through the Access Bridge. GTP is an abstract protocol and, for this reason, needs to be mapped onto some concrete protocol. GTP is designed in such a way that a concrete tunneling protocol is provided as an adaptation layer between GTP and the real transport layer protocol (see Figure 2). Thereby, in order to bring wireless CORBA to some particular wireless network, the adaptation layer between GTP and some protocol of this network has to be implemented.

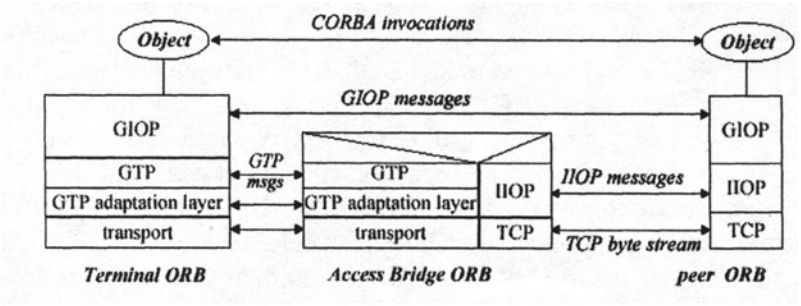

Figure 2. GIOP Tunneling Protocol Architecture [1]

4. BLUETOOTH AND WIRELESS CORBA

A Bluetooth device, which has an interface to some network and which acts as a gateway to that network, can act as an access point for other Bluetooth devices. Hence, it should be possible, at least in theory, to use Bluetooth access points as wireless CORBA access bridges and access CORBA services from Bluetooth data terminals. Bluetooth also has the

capability of forming other kinds of networks, ad hoc networks, that are not necessarily connected to other networks and that can be formed anywhere, anytime, using the proper devices. However, ad hoc networks are beyond the scope of this paper.

As was mentioned in chapter 2, the wireless network should fulfill certain requirements in order to enable wireless CORBA to function on it. This chapter discusses how well Bluetooth meets these requirements and what kind of problems exist when using wireless CORBA on Bluetooth access networks.

4.1 The Problem of Seven Active Slaves

Bluetooth is distinguished from other well-known wireless communication technologies (like Wireless LAN or GPRS) by the very strict limitations it imposes on the number of simultaneous connections it accepts. In a Bluetooth, a strict master-slave scheme is employed where slave devices can communicate with each other only through the master device. The master and slaves form *a piconet*, where the master can maintain up to seven *active slaves* connected. An active slave is a Bluetooth device which is able to communicate and transfer data with the master. More slaves may exist in a piconet but only if they are in a low-power mode where they are not able to transfer data with the master.

The most convenient architecture for Bluetooth access networks is one where the Bluetooth access point acts as a master and the mobile terminals accessing it act as slaves (see Figure 3). The seven active slaves restriction causes major problems in this kind of architecture whenever a new data terminal desires to connect to an access point which is already fully reserved, having seven slaves connected already. In this case, the Access Point refuses to serve the most recent Bluetooth device until one of the already connected data terminals releases its connection. The worst scenario is one in which the access point is full and slaves, which may not necessarily perform any data transfer, have already reserved the active places in the piconet.

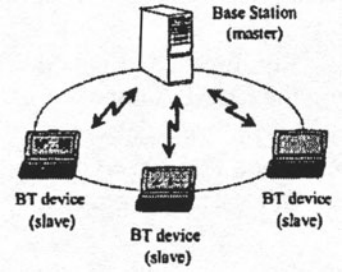

Figure 3. Bluetooth architecture [3]

It is obvious that the problem of the seven active slaves decreases the usability of certain access points. However, in practice, there always exists some kind of limit to how many clients an access point can serve. The access point shares its bandwidth among the connected slaves and thus, the number of connected slaves cannot be much greater than seven if the proper connection is to be guaranteed to all slaves. The problem is, therefore, not the small number of connected slaves but rather the existence of data terminals that are connected to the access point or that have reserved active places from the piconet without having a reason for doing so. The problem can be solved with the help of *park mode*, which is a power-save mode where a slave in a piconet remains synchronized to the master but is not allowed to transfer data. Thereby, at a particular moment, when there is no data traffic, the idle active slave should be switched to the park mode in order for an active slave place to be released from the piconet for devices that may really need it. Switching to the park mode and back again can be done at any time by a decision from the master. A slave can also leave a request to be switched to one or the other mode. Whenever a parked slave needs to transfer data again or if the master receives some data targeted to a parked slave, the slave can be switched back to the active mode. Of course, in order to switch back to the active mode, there has to be room in the piconet for one more active slave. If there is no room, some other active slave can be switched to the park mode. In a case where the piconet is full and more than seven slaves need to transfer data, some kind of prioritisation can be used for managing the switching of slaves to the park mode and back.

4.2 Wireless CORBA Handoff

Wireless CORBA specifies methods for handoff, moving from an access point to another without losing the CORBA connection. There exist two kinds of handoffs: 1) backward handoff is a method where the connection between the mobile terminal and the access points are not dropped, 2) forward handoff is a method where the connection is recovered after a sudden loss of connection. Generally, handoff can happen because of following reasons:

1. a mobile terminal physically moves and changes the current access point to another without losing the CORBA connection (backward handoff);
2. a mobile terminal is switched from the current access point to another, without losing the connection, by a server that distributes data traffic between access points in order to provide better bandwidth usage (backward handoff);
3. for some special reason, the user of the Bluetooth mobile terminal wants to change the current access point to another without losing the

connection (backward handoff, an example of which is discussed in chapter 4.5);

4. a mobile terminal unexpectedly loses the connection to the current access point and the connection is re-established (forward handoff).

Implementing forward handoff for Bluetooth should be quite easy because it involves simply re-establishing a connection that has been lost. The backward handoff process causes more problems. Wireless CORBA specifies that a connection to the new Access Bridge has to be established before the connection to the old access point can be released. However, if the terminal cannot maintain simultaneous transport connectivity to the old and new Access Bridge, the so-called "alternative handoff procedure" can be used. The alternative handoff procedure first releases the connection to the old Access Bridge and only afterwards establishes a connection to the new Access Bridge. It is obvious that the performance in the latter approach is much worse than in the former. The requirement of "real" backward handoff to have simultaneous connections to both the new and old access points makes its implementation impossible for existing Bluetooth devices. The reason for this is that they do not provide support for *scatternet* which allows Bluetooth slaves to be connected to more than one master at the same time. However, while the support for scatternet is awaited, the alternative implementation of backward handoff can be used without problems, but of course at the price of performance.

4.3 Starting the Backward Handoff Procedure

Another, much more difficult problem, is to recognize when the backward handoff process should take place. Bluetooth was originally designed to simply replace cables, and there was no intention to make it work as a real wireless network and provide mobility services etc. This is the most probable reason why efficient mechanisms have not been incorporated into Bluetooth in order to determine its location inside the piconet. However, for backward handoff, location information is essential because without this information, it is impossible to know when the device should start the process of changing the access point. The device may be anywhere inside the piconet - on the edge of the piconet or right beside the master device. The only information that can be obtained is that the connection has been lost which is, for instance, the case when the device moves out of the piconet.

In GSM, the decision of changing the base station is made upon strength of the radio signal. The Bluetooth specification does not define any methods for measuring the strength of the radio signal but there are some methods that can be used. To achieve a hint of the location of the terminal, the so-called inquiry process (explained in chapter 4.4) can be run periodically.

However, inquiry is a very heavy process that takes quite a long time and only provides information on what devices exist in the vicinity, not telling, for instance, the mobile terminal to which access point it should connect to or how far the device is from different access points. Another, and much more convenient possibility, would be to use the Bluetooth method "strength of link" (also known as "link quality"). This method does not measure the radio signal strength, but calculates a value on the basis of the number of access errors and retransmissions during data transfer. The greater the distance of the mobile terminal to the master, the greater the number of errors and retransmissions is and thereby, the link quality value can be used to detect when a terminal is moving outside the piconet and there is a need to initiate the handoff process.

The measurement of the strength of a link can be implemented in two different ways depending on whether the mobile terminal or the access point is performing the measurement process. However, measuring the strength of a link only gives a hint of the distance between the master and the slave, not providing accurate coordinates. It is impossible to know in which direction the device is moving and to which access point the device should try to connect. There may very well be another reason, other than the distance, for the increase in the number of errors (for example, the interference could be caused by physical obstacles or other devices). Nevertheless, "strength of link" seems to be the only feasible way of obtaining information on mobility if measuring the strength of the radio signal is not possible.

4.4 Connection Establishment to a New Access Point in Backward Handoff

Bluetooth incorporates special mechanisms for establishing connections with other devices. Inquiry is a process where a Bluetooth device searches for other Bluetooth devices in the area and obtains their Bluetooth addresses. Paging is a process where the connection to the device that was found in inquiry is established. In backward handoff, when there is a need to establish a connection between the new access point and the mobile terminal, the following procedure is followed:

1. the mobile terminal does not know if there exist any other access points in the area and hence, the inquiry process has to be run by the mobile terminal in order to find new access points and obtain their Bluetooth addresses;
2. after an appropriate access point is found and the corresponding address received, the mobile terminal can connect to it by running the paging process.

Alternatively, the access point can run the inquiry and paging processes and establish the connection to the mobile terminal. In any case, the problem in the Bluetooth procedure for connection establishment is the inquiry process, which is very heavy and thus consumes substantial amounts of time. As can be seen from Table 1, the inquiry and paging processes together take approximately between 4.28 and 6.28 seconds on average, but can, in the worst case, take up to tens of seconds. In backward handoff, where the mobile terminal moves between Bluetooth access points that are within a range of 10 meters, a delay of between 4 and 6 seconds for Bluetooth connection establishment is unacceptably long. In addition, the wireless CORBA handoff process will consume some time. At worst, the delay can be over 30 seconds, which is totally unacceptable.

Table 1. The time taken to complete the inquiry/paging procedures [4]

Operation type	Minimum Time	Average Time	Maximum Time
Inquiry	0.00125	3-5	10.24 - 30.72
Paging	0.0025	1.28	2.56
Total (Paging + inquiry)	0.00375	4.28 - 6.28	12.80 - 33.28

However, if the address of the new access point is somehow known in advance, there is no need to run the time-consuming inquiry process. If the address is known, only the paging process is required for establishing the connection, which takes only 1.28 seconds on average and, even at worst, only 2.56 seconds (Table 1). The address of the new access point can be known in advance if, for example, the mobile terminal receives the address information from the current access point. If the access points are responsible for establishing the connection, the current access point delivers the address of the mobile terminal to the next possible access points. These access points then run the paging procedures, and the connection is established between the mobile terminal and the access point in the area of which the mobile terminal happens to be. Once again, only paging is needed. The point of delivering the address information is that the inquiry process only has to be executed once when the connection between the first access point and the mobile terminal is established.

Another disadvantage of the inquiry process is the fact that during this process, devices that already have a Bluetooth connection to the inquiring device are not able to communicate with it [3]. This is the main reason why the inquiry process should not be run by any access points, and is in actual fact a major disadvantage for the whole Bluetooth technology. For example, let assume that there exists a mobile phone with a Bluetooth headset and a laptop that has a Bluetooth chip. Now, if the mobile phone and the headset are used for a telephone call, and during the call the mobile phone runs an

inquiry to find the laptop, there will be a break in communication between the mobile phone and the headset until the inquiry process is terminated.

4.5 Backward Handoff between Different Technologies

Backward handoff between different technologies may be even more important than normal backward handoff between Bluetooth access points. For example, a user may currently be using a GRPS connection but would like to switch to a Bluetooth connection when they arrive in the vicinity of a Bluetooth access point (perhaps because Bluetooth is cost-free). Vice versa, a user using a Bluetooth connection may want to switch to GPRS if they know that they will soon move out of the range of Bluetooth and risk losing the connection.

Implementing backward handoff between different technologies should be quite easy, at least if the user makes the decision to switch from one technology to another (manual handoff). In such a case, all problems related to automatic handoff can be forgotten (Chapter 4.3). The user is, most probably, really responsible for making the decision to switch between technologies, at least if switching means alternating between cost-free services and services that carry charges and vice-versa (for example, from Bluetooth to GPRS). The process can also be automated, although the user should always be aware of when the new connection will carry a charge and when it will not.

4.6 Connecting to the Correct Access Points

An area in which a device moves can contain several Bluetooth access points all of which may not necessarily provide access to wireless CORBA services and there must, therefore, be some way of recognizing the correct access points. Bluetooth provides the mechanisms to request information on what kinds of services exist on a specific Bluetooth device, although the problem is that the connection to the device first has to be established. This information cannot, therefore, be obtained through inquiry calls, for instance.

Connection establishment takes time and if the connected device does not provide the required services, even more time is wasted when connecting to another access point. Nevertheless, as was discussed in previous chapters, connection establishing can be done in order for the inquiry process to be carried out only once, when the first connection is established. Afterwards, the addresses of new access points are always obtained from the current access point, so that only the paging process needs to be run. This sort of an approach also solves the problem of connecting to the correct access points because it is possible to transfer only the addresses of those access points

that provide some particular services such as, in this case, access to wireless CORBA services.

5. CONCLUSIONS AND FURTHER WORK

The disadvantages of Bluetooth technology impede it from fully meeting the requirements set by wireless CORBA. Bluetooth, in its present form, enables the use of wireless CORBA at some level, while the proper use of CORBA is not possible. The usability of Bluetooth access points is low, connection establishment mechanisms are slow, correct access points cannot be recognized and the support for automatic handoff cannot be implemented. However, as the research carried out by the authors has shown, these problems can be avoided. Switching active idle slaves to the park mode solves the problem of the seven active slaves. Obtaining the address of the new access point from the current access point speeds up the connection establishment process and the time-consuming inquiry process need not be run. This approach also solves the problem of recognizing the correct access points. The strength of the link can be used to obtain the impulse for starting the backward handoff process and automatic handoff can be implemented. Thus, these kinds of extensions to Bluetooth allow the proper use of wireless CORBA over Bluetooth.

In our proximate work, we will study which protocol in the Bluetooth protocol stack can be used for tunneling and will specify the adaptation layer between that protocol and GTP. Furthermore, we carry out a prototype implementation of tunneling and test our work against real wireless CORBA implementation.

REFERENCES

[1] Object Management Group, "Wireless Access and Terminal Mobility in CORBA specification", June 2001

[2] E. Amir, H. Balakrishnan, S. Seshan, R. Katz, "Efficient TCP over Networks with Wireless Links", May 1995.

[3] S. Baatz, M. Frank, R. Göpffarth, D. Kaastkine, P. Martini, M. Schetelig, A.Vilavaara. "Handoff Support for Mobility with IP over Bluetooth", 2000

[4] Palowireless - Bluetooth Resource Center, "Time Taken to complete Inquiry/Paging Procedures".

[5] http://www.palowireless.com/infotooth/knowbase/baseband/99.asp [Accessed 26 Sep 2001]