Middleware Services over Satellite Networks: A Survey of Issues

Steven Gordon and Jonathan Billington

Cooperative Research Centre for Satellite Systems

University of South Australia

Mawson Lakes SA 5095, Australia

e-mail: {sgordon,jb}@spri.levels.unisa.edu.au

Abstract

Middleware promises seamless integration of heterogeneous networks and computing/telecommunications environments by providing distributed services to user applications. With Low Earth Orbit satellite networks being deployed as communications infrastructure, it is important that middleware can operate efficiently over these networks. For applications to have good performance, efficient and robust communications between middleware objects is mandatory. Many protocols have been proposed to provide mobility, and to improve data transport over the restrictions imposed by satellite links. This paper provides a survey of the issues and potential solutions, with the aim of stimulating further research in this developing area.

1 Introduction

The use of computer networks, like the Internet, for both business and personal use has become an important part of everyday life. This will increase in the future with wireless technologies providing the infrastructure to access these networks at anytime and from anywhere in the world. In particular, Low Earth Orbit (LEO) satellite networks are being deployed to provide coverage of wide, and often isolated areas (e.g. Iridium (`www.iridium.com`), Teledesic (`www.teledesic.com`)). As mobile devices such as laptop computers, cellular phones and Personal Digital Assistants (PDAs) become more powerful and sophisticated, users increasingly expect to be able to perform similar operations as they can with their desktop devices (e.g. Web browsing, running multimedia applications). However in a widespread heterogeneous environment the problem of interoperability between applications arises. To cope with this, the use of middleware services is necessary.

Middleware [1] provides distributed services that support the use of applications in heterogeneous computing environments. These services provide transparency of the underlying operating systems, hardware and networks allowing developers to concentrate on the business logic of their applications. Middleware services can be viewed as a software layer between platforms and applications (figure 1).

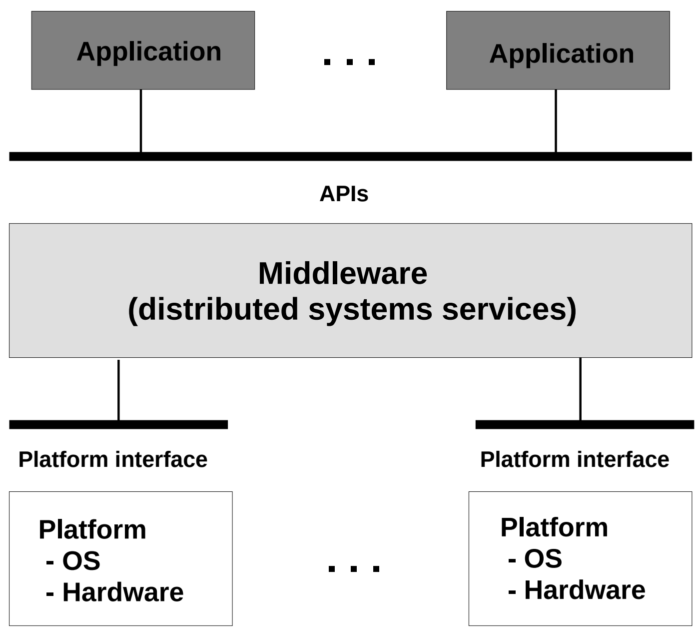

Figure 1: Middleware Layer [1]

Some examples of middleware services are: security, naming, maths functions, trading, transaction management, accounting and data interchange. The trading service, which offers the location of other services and objects, is of particular interest in a mobile computing environment. It may be used to locate, for example, nearby hospitals, restaurants or petrol stations, when travelling in unfamiliar regions.

Middleware will play an important role in the provision of new applications and services over LEO satellite networks to mobile users. This paper discusses the issues that may affect the development and operation of middleware services in satellite networks. Firstly middleware and trading are introduced (Section 2). The limitations and characteristics of LEO satellite networks, listed in Section 3, will affect the performance of all systems required to operate over satellites, including middleware services. Section 4 discusses how these affect middleware, and the efforts being made to solve the problems. Section 5 finishes with some concluding remarks.

2 Middleware

2.1 Types of Middleware

Generally, middleware can be divided into five different types [2] based on the services provided and where they are used:

1. Database middleware.
2. Transaction Processing (TP) Monitors
3. Message Oriented Middleware (MOM)
4. Object Request Brokers (ORBs)
5. Remote Procedure Call (RPC) based middleware

Both TP monitors and database middleware are used for accessing data. The other three types can be considered as communication middleware. They are concerned with providing services for communication between components in a distributed system. MOM provides asynchronous program-to-program data exchange. RPC based middleware and ORBs provide communication between components of a program (a lower level service than MOM). The main difference between the two is that RPC is a procedural approach whereas ORBs are an object-based approach. Leading technologies for the more dominant of the two, ORBs, are examined in more detail in the following sections.

2.1.1 CORBA

The Object Management Group (OMG) (`www.omg.org`), a consortium of software vendors, developers and end users, have specified an architecture to enable applications to interoperate in a heterogeneous, distributed computing environment. This is known as the Object Management Architecture (OMA) [3] (figure 2).

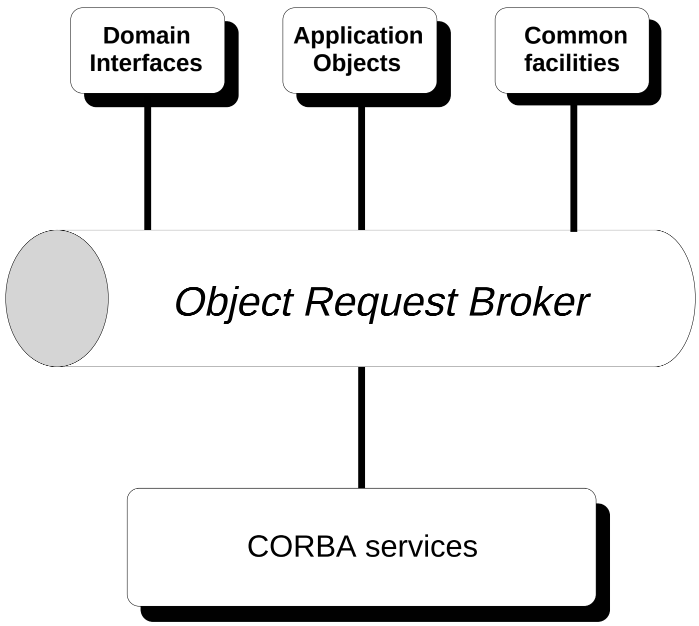

Figure 2: Object Management Architecture Reference Model

The OMA consists of components that define objects and their interfaces. It is centred around an Object Request Broker (ORB) which provides a communication infrastructure for objects. The ORB delivers requests to objects and returns any responses, while hiding the location, implementation, execution state and communication mechanisms of the objects. The Common Object Request Broker Architecture (CORBA) [4] is the technology OMG has specified for the ORB. The other parts of the OMA are:

- CORBA services - services that are needed for implementing and manipulating objects (e.g. naming, trading, security, concurrency control, event notification),
- Application Objects - objects that are application specific (OMA do not plan to develop a standard for this component),
- Domain Interfaces - facilities for objects specific to a domain e.g. telecommunications, healthcare, manufacturing,

- Common facilities - facilities for User Interface, Information Management, System Management, and Task Management (e.g. support for printing, email, compound documents).

These services and facilities are accessed by each other via the ORB. For example, an application object in search of an object with a known name, can, via the ORB, submit a request to the naming service and receive a response with the named object's reference identifier (i.e. location).

As CORBA is a specification there will be a large variety of CORBA-compliant ORBs. OMG have specified protocols for interoperability between ORBs. The General Inter-ORB Protocol (GIOP) [4] is a basic protocol designed to be easily implemented on most transport systems. A specific variant of GIOP is the Internet Inter-ORB Protocol (IIOP) [4]. Using this protocol, ORBs can communicate across the Internet (i.e. using TCP/IP). Other variants of GIOP are Environment Specific Inter-ORB Protocols (ESIOP) [4]. CORBA ORBs must comply with both GIOP and IIOP.

With recent efforts to increase interoperability with Java [5], CORBA is a strong competitor in the middleware market.

2.1.2 DCOM

Distributed Component Object Model (DCOM) [6] is middleware that provides similar functionality to CORBA. DCOM adds distributed services to the Component Object Model which provided an infrastructure for communication between objects in a Microsoft Windows based computer. Therefore, being a Windows-first solution, some DCOM capabilities are limited [5]. However, with the large number of Windows products, and ports being made to other operating systems, DCOM is highly competitive with CORBA/Java. Approaches to bridge the two technologies are already in development [4].

2.2 Trading

One of the middleware services offered is a trader. A trader is used to locate objects in an environment (see figure 3). It acts like a yellow pages directory where objects advertise their services (1), and other objects use the trader to find them (2 and 3). If found, the service can be invoked (4 and 5).

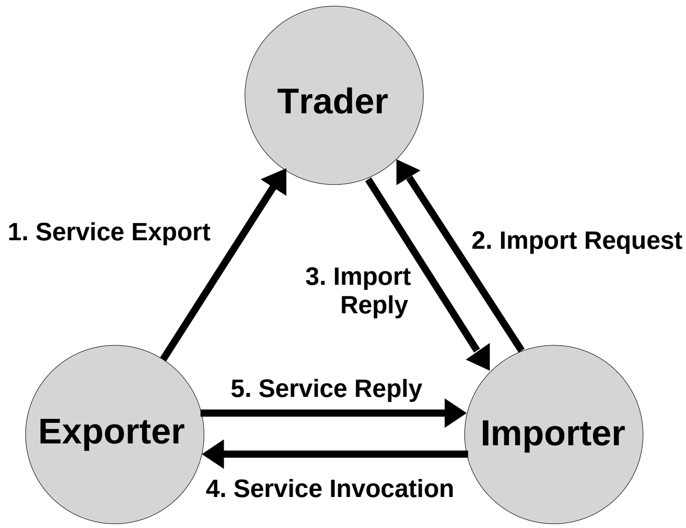

Figure 3: Trading Environment

Traders can interwork with each other to increase the offer space for services. However, this requires further policies and protocols to maintain the links between traders and provide efficient search strategies.

The applications of traders include service provision and management in telecommunications environments [7], load balancing in operating systems [8] and object location in multimedia applications [9]. Additionally, in satellite networks, traders may be used for locating commercial services such as petrol stations, hospitals, restaurants etc.

3 LEO Satellites

To provide wide (almost global) coverage areas, constellations of LEO satellites are being deployed as a communications infrastructure. These networks will not only service remote areas, but also compete with existing wired and wireless networks. There are many advantages and disadvantages of using wireless communications instead of wired technologies [10]. These are associated with the services offered (e.g. mobility, portability), performance (e.g. error rates), regulation (e.g. allocation of spectrum), and financial issues (e.g. cost of setting up infrastructure in, for example, a building without adequate wiring). In general these are also true for satellite communications. However, there are some significant differences between LEO satellite communications and land-based wireless communications [11]. This section identifies the characteristics of LEO satellites that differentiate them from geostationary satellites, terrestrial wireless and fixed communication technologies, and that may have an affect on the way middleware is developed and operates.

- *High error rates.* Error rates are typically higher in satellite communications than other fixed mediums because of the power loss in transmission (e.g. free space loss, antenna losses) and interference (e.g. multipath fading, shadowing).

- *Asymmetric channels.* For satellite communications it is common to have channel capacities much higher for downlinks (satellite to earth) than uplinks. This is because of various tradeoffs between power, cost and mass in transmitters.

- *Disconnections.* Due to high error rates or satellites not being in the line of sight of an earth terminal there may be times when the earth terminal is not connected to the network.

- *Handover.* In LEO satellite constellations, handover between satellites is required to maintain connections due to movement of the satellites with respect to the ground. This is in addition to handovers due to terminal mobility (which is common in terrestrial cellular networks). Ideally, the changes in routing of the connection would be transparent to the user, whether they are fixed or mobile.

- *Delay.* The propagation delay between LEO satellites and a ground station is of the order of milliseconds which may restrict the operation of some systems, especially real-time communications.

- *Limited bandwidth.* The bandwidth available (link capacity) is lower when compared with wired communications (due to regulations on frequency assignment and tradeoffs with terminal power and error rates). For example, most LEO satellite networks offer 2-20 kbit/s (with the exception of Teledesic which aims to provide 16-2048 kbit/s, although the higher rates only for fixed terminals), whereas data rates of 28 kbit/s (dialup modem) to 10's, even 100's, of Mbit/s are common in wired networks. Hence, the use of the bandwidth must be more efficient.

- *Security.* The security of wireless communications in general is not as good as fixed networks because interception of the signals is much easier. Encryption and scrambling mechanisms are often used to combat this.

4 Middleware over Satellite Networks

4.1 The Problems

The limitations of LEO satellite networks will affect the operation of middleware. One major reason is that middleware makes use of the Internet for communication (e.g. DCOM, CORBA IIOP). However, there are problems with current Internet protocols used over wireless mediums, and especially satellites [12]. These are outlined below:

- TCP assumes that packets are lost only due to network congestion. When this occurs the congestion control mechanism reduces the window size and hence transmission rate. However, on a satellite link, packets are also lost due to the high error rates. In this case TCP will assume network congestion and reduce the transmission rate. This is an undesirable response to transmission errors.

- TCP/IP considers a broken link (i.e. no response within a given time) to be a failure. The connection is closed and existing operations are stopped. Disconnection is not uncommon in satellite networks. A more efficient way of handling them would be useful in TCP/IP.

- The size of headers used in TCP/IP is of concern because of limited bandwidth. Removal of unnecessary data or compression could be useful for use in satellite networks.

Although improvements to TCP/IP are being made to support satellite communications (see Section 4.2), further changes to middleware will be needed. Some examples are:

- Enhancing the performance of middleware services. Although extensions to TCP/IP might improve performance compared to normal TCP/IP over a satellite link, it is unlikely that the performance will reach that of a fixed network for which current middleware services are designed.

- Designing middleware services and infrastructures that are specific to satellite environments. For example, the organisation of interworking traders is critical in satellite networks because the delays when traversing links may be large. Determining where to locate traders (in the mobile host, base station or fixed host) is an open problem.

4.2 Proposed Solutions

4.2.1 Wireless Access

Many extensions of TCP/IP have been proposed to alleviate the above mentioned problems. Nearly all of the solutions can be categorised using one of three approaches [13]: removing the wireless errors using a lower level protocol (e.g. Snoop [14]); using two TCP connections, a normal connection between the fixed host and base station, and an improved transport protocol over the wireless link (e.g. I-TCP [15]); and enhancing the end-to-end TCP connection (e.g. using Selective Acknowledgements [16] or Explicit Loss Notification [13]).

SCPS-TP (Space Communications Protocol Standards - Transport Protocol) [12] is a set of extensions to TCP to support space communications (from LEO satellites to deep space probes). It could also be used as the improved transport protocol in, for example, I-TCP. Another alternative for an improved transport protocol for a satellite link is Satellite Transport Protocol (STP) [17].

The Wireless Application Protocol (WAP) [18] is an architecture that uses a split connection approach to provide wireless Internet services to handheld devices (e.g. PDA's, cellular phones). The architecture, consisting of a set of protocols and an application framework, is defined by the WAP Forum (`www.wapforum.org`). With industry support behind it, WAP is a contender to become a defacto standard for providing Internet services to handheld devices. This will also influence middleware services in satellite networks.

The approach of using a wireless specific protocol over the wireless segment and TCP/IP for the fixed segment has been used in the ACTS DOLMEN project [19]. This is a specific example of extensions of middleware for the wireless environment. A Lightweight IOP (LW-IOP) is used between an access point in the mobile terminal and an access point to the fixed network. The protocol is converted to IIOP at each of the access points so that CORBA ORBs can utilise IIOP in the terminal and fixed network. LW-IOP provides the efficiency (e.g. low message overheads, string caching) and functionality (e.g. coping with disconnections) needed for wireless access.

Within OMG the Telecom Domain Task Force are investigating the support needed, if any, for wireless access and mobility in CORBA [20]. The interest from OMG and responders to the initial Request For Information (RFI) indicate more work is needed (e.g. identifying the services to be provided by CORBA and an ESIOP and those to be provided by the networking layer, and their interactions) to provide complete answers.

4.2.2 Mobility

The effects of mobility on middleware has attracted some research [21, 22], the most notable within the ANSA project [23]. Their approach was to handle the effects of mobility (e.g. fluctuating quality of service in the communications network) by providing applications with knowledge about the network [22]. This was also applied to Wide Area Networks [24]. This is the opposite of what other platforms try to do, which is hide the network characteristics from applications. ANSAware has been extended to support mobile applications, in particular multimedia applications [25]. This was done by modifying the RPC mechanism to take into account changes in quality of service. These same concepts have also been applied to a CORBA environment [26].

4.2.3 Performance

The performance of middleware in both wireless and wired environments needs continual improvement if it is to be used on high speed networks, especially for mission/life critical applications [27, 28]. Some overheads of ORBs and RPC toolkits when compared to C/C++ socket communication include inefficient data copying and memory management and excessive use of control information in messages. Although it is expected that the lower level operations (C/C++ sockets) will be more efficient, without improvements (i.e. by allowing ORBs to make trade-offs between functionality and performance) the benefits of using CORBA and RPC may not outweigh their slowness in high speed networks.

5 Conclusion

LEO satellite networks will enable advanced mobile computing applications across wide and remote areas, as well as competing with other technologies where infrastructure exists. To support the development of these mobile applications, middleware such as CORBA must operate effectively over satellite networks. This paper has presented the issues that may affect the operation of middleware services over satellite networks.

Transport layer problems using satellites are a common problem that will affect middleware. Many solutions have been proposed but it is still not clear how much knowledge middleware services should have about the satellite link. These issues are beginning to be tackled by OMG. Other middleware issues, such as the organisation of interworking traders, also need to be considered.

Research on middleware over satellite networks is limited. For trading services, it is almost non-existent. Given the benefits of rapid deployment of advanced mobile computing applications in heterogeneous networks, the interest in middleware services over satellite networks will expand. Further investigation and resolution of the issues presented in this paper is urgently required.

Acknowledgements

This work was carried out with financial support from the Commonwealth of Australia through the Cooperative Research Centres Program.

References

[1] P. A. Bernstein, “Middleware: A model for distributed systems services,” *Communications of the ACM*, vol. 39, no. 2, pp. 86–98, Feb. 1996.

[2] International Systems Group, “Middleware - the essential component for enterprise client/server applications,” White Paper, Feb. 1997.

[3] Object Management Group, *A Discussion of the Object Management Architecture*, Object Management Group, Framingham, MA, Jan. 1997.

[4] Object Management Group, *The Common Object Request Broker: Architecture and Specification, Revision 2.2*, Object Management Group, Framingham, MA, Feb. 1998, OMG Document: `formal/98-07-01`, available via `www.omg.org`.

[5] R. Orfali and D. Harkey, *Client/Server Programming with Java and CORBA*, John Wiley and Sons, New York, second edition, 1998.

[6] Microsoft, “DCOM technical overview,” White Paper, 1996.

[7] J. W. Hong and J.-T. Park, “Supporting distributed management in a telecommunications management network (TMN) environment,” in *Proceedings of the IEEE Second International Workshop on Systems Management*, Toronto, Canada, June 1996, pp. 148–157, IEEE Computer Society.

[8] A. W. Pratten, J. W. Hong, J. M. Bennett, M. A. Bauer, and H. Lutfiyya, “Design and implementation of a trader-based resource management system,” in *Proceedings of the 1994 CAS Conference*, Toronto, Canada, Oct. 1994, pp. 130–141.

[9] J. Bates, D. Halls, and J. Bacon, “A framework to support mobile users of multimedia applications,” *Mobile Networks and Applications*, vol. 1, no. 4, pp. 409–419, 1996.

[10] A. S. Tanenbaum, *Computer Networks*, Prentice-Hall, Englewood Cliffs, N.J., 3rd edition, 1996.

[11] T. Lodgson, *Mobile Communication Satellites*, McGraw-Hill, New York, 1995.

[12] R. C. Durst, G. J. Miller, and E. J. Travis, “TCP extensions for space communications,” *Wireless Networks*, vol. 3, no. 5, pp. 389–403, Oct. 1997.

[13] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, Dec. 1997.

[14] H. Balakrishnan, S. Seshan, and R. Katz, “Improving reliable transport and handoff performance in cellular wireless networks,” *Wireless Networks*, vol. 1, no. 4, pp. 469–481, Dec. 1995.

[15] A. Bakre and B. R. Badrinath, “Indirect transport layer protocols for mobile wireless environment,” in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds., pp. 229–252. Kluwer Academic Publishers, Boston, MA, 1996.

[16] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP selective acknowledgement options,” IETF RFC 2018. URL: `http://www.ietf.org/rfc/rfc2018.txt`, Oct. 1996.

[17] T. R. Henderson and R. H. Katz, “Transport protocols for Internet-compatible satellite networks,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 2, pp. 326–344, Feb. 1999.

[18] WAP Forum, *Wireless Application Protocol Architecture Specification*, Available at: `http://www.wapforum.org/what/technical/arch-30-apr-98.pdf`, 30 Apr. 1998.

[19] M. Liljeberg, K. Raatikainen, M. Evans, S. Furnell, N. Maumon, E. Veldkamp, B. Wind, and S. Trigila, “Using CORBA to support terminal mobility,” in *Proceedings of TINA’97: Global Convergence of Telecommunications and Distributed Object Computing*, Santiago, Chile, 17-20 Nov. 1997, pp. 59–67, IEEE Computer Society.

[20] Object Management Group, *RFI: Supporting Wireless Access and Mobility in CORBA*, Object Management Group, Framingham, MA, June 11, 1998, OMG Document: `telecom/98-06-04`, available via `www.omg.org`.

[21] N. Diehl, D. Grill, A. Held, R. Kroh, T. Reigber, and T. Ziegert, “System integration for mobile computing and service mobility,” in *Distributed Platforms*, A. Schill, C. Mittasch, O. Spaniol, and C. Popien, Eds., pp. 44–56. Chapman & Hall, London, U.K., 1996.

[22] N. Davies, “The impact of mobility on distributed systems platforms,” in *Distributed Platforms*, A. Schill, C. Mittasch, O. Spaniol, and C. Popien, Eds., pp. 18–25. Chapman & Hall, London, U.K., 1996.

[23] R. van der Linden, “An overview of ANSA,” Tech. Rep. APM.1000.01, Architecture Projects Management Ltd., Cambridge, UK, July 15, 1993.

[24] D. E. Bakken, R. E. Schantz, and J. A. Zinky, "QoS issues for wide-area CORBA-based object systems," in *Proceedings of Second Workshop on Object-Oriented Real-Time Dependable Systems (WORDS'96)*, Laguna Beach, CA, 1-2 Feb. 1996, pp. 110–112, IEEE Press.

[25] A. J. Friday, *Infrastructure Support for Adaptive Mobile Applications*, Ph.D. thesis, Computing Department, Lancaster University, England, Sept. 1996.

[26] G. Blair, G. Coulson, N. Davies, P. Robin, and T. Fitzpatrick, "Adaptive middleware for mobile multimedia applications," in *Network and Operating System Support for Digital Audio and Video (NOSSDAV '97)*, St Louis, MO, 19-21 May 1997, pp. 245–254.

[27] A. Gokhale and D. C. Schmidt, "Measuring the performance of communication middleware on high-speed networks," in *Proceedings of ACM SIGCOMM*, Stanford, CA, 26-30 Aug. 1996, pp. 306–317.

[28] D. C. Schmidt, A. S. Gokhale, T. H. Harrison, and G. Parulkar, "A high-performance end system architecture for real-time CORBA," *IEEE Communications Magazine*, vol. 35, no. 2, pp. 72–77, Feb. 1997.