MIDDLEWARE SOLUTION FOR ALL IP NETWORKS

K E E Raatikainen

Nokia Research Center, Software Technology Laboratory

Abstract. *Telecommunication networks will be based on the Internet Protocol or IP in the future. As a network layer protocol the IP protocol alone is not sufficient; not even with suitable Internet transport protocols such as TCP, UDP, and RTP. In the Internet community, that is the Internet Engineering Task Force or IETF, a larger number of activities is going on to specify new protocols and Application Programming Interfaces or APIs to provide a complete networking solution.*

This paper examines the implications of all IP-networks on middleware. A special emphasis is on interoperability of specifications defined in various standardization bodies, for a, and consortia. In particular, we discuss the danger of interpreting "all IP" as specified in the IETF since then a huge amount of work, well-established specifications and mature software products is not employed

INTRODUCTION

The current trend in developing forthcoming telecommunication networks is to utilize Internet protocols. An immediate implication is that IP is the layer 3 protocol and that the addresses are IP addresses. However, this is not sufficient. Other solutions—both above and below the IP protocol—are also needed to meet the requirements for the next generations of telecommunication networks. Issues under study in the Internet community and in various standardization bodies, forums and consortia of telecommunications include mobility, Quality-of-Service, security, management of networks and services, discovery, ad-hoc networking and dynamic configuration, geospatial location.

Another significant trend is the requirement of ever-faster service development and deployment. The immediate implication has been the introduction of various service/application frameworks/platforms. Middleware is a widely used term to denote a set of generic services above the operating system. Although the term is popular, there is no consensus of a definition [1].

Typical middleware services include directory, trading and brokerage services for discovery, transactions, persistent repositories, different transparencies such as location transparency and failure transparency. Examples of middleware include Common Object Request Broker Architecture (CORBA) [2], Java 2™ Enterprise and Micro Editions (J2EE and J2ME) [3, 4], Distributed Common Object Model (DCOM) [5], and Wireless Application Environment (WAE) [6]. Characteristically, the competing middleware specifications provide many similar but slightly different services. In order to overcome the problems due to different specifications, the Parlay Group [7] has specified a set of APIs that can be implemented in CORBA, Java and DCOM environments.

The rest of the paper is organized as follows. We start by briefly outlining an execution environment of future distributed mobile applications derived from application requirements. In the section of Internet Protocols and Middleware we discuss programming environments. Finally, in the section of Research Issues we summarize developments in Internet protocols and state requirements for interoperability between middleware solutions.

APPLICATION REQUIREMENTS

Figure 1 depicts a highly abstracted vision of how a service application is distributed among various application servers, network elements and terminal or end-user systems. It should be noted that, for simplicity, the figure only shows a single terminal device although multi-party applications will be much more important and challenging than one-party applications such as information browsing. In addition, we must also be ready to cope with end-user systems based on body area networks and home communication systems.

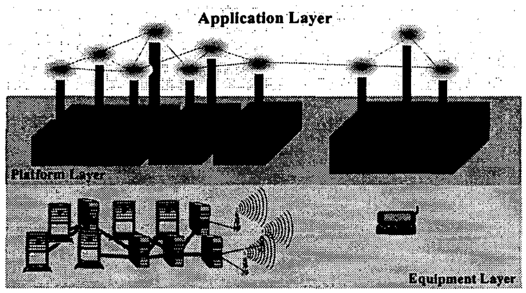

Figure 1: Partitioning and Distribution of Application Logic

The execution environments or the platform layer consist of middleware, operating systems and protocol stacks that should support fast service development and deployment. The platforms should make it easy to divide the application logic into co-operating parts—someone may call them components, to distribute and configure these components as well as to redistribute and reconfigure them. Additional requirements of future applications for nomadic users include adaptability to changes in the execution and communication capabilities, efficient use of available communication resources, dynamic configuration of end-user systems as well as ultimate robustness, high availability and stringent fault-tolerance.

Adaptability

Adaptability is one of the key research areas in nomadic computing. The basic principle of adaptability is simple. When the circumstances change, then the behavior of an application changes according to the desires of a user—or more precisely according to principles ascribed to her.

Learning the wishes and desires of a user is a crucial part of adaptability. Another crucial problem is the size and computational complexity of the knowledge base. Most probably the knowledge needs to be partitioned so that each subset is small enough. The partitioning, however, is not alone sufficient. The models presenting the subsets of the knowledge must be combined in different ways for different purposes.

Adaptability cannot only be reactive. When the battery dies or the connectivity breaks, many actions are impossible. However, something could have been done beforehand. Therefore, adaptability must also be proactive, which, in turn, requires predictability of the near future. An important question in predictions is to distinguish between the situations in which the user behavior seems to be predictable and those being unpredictable.

Wireless Communication

As a communication channel, air is problematic. In the last ten years the progress in coding has significantly increased the capacity of wired channels. Unfortunately these fruits cannot fully be utilized on wireless channels. The applied coding is always a compromise between information density and redundancy providing robustness against interference. The basic problem with wireless links is instability in the sense that the level of interference varies in time and place, and according to environmental conditions.

It is very often said that the speed of 2 Mbits per second, which is assumed to be available in the 3G, will be fast enough. However, in the history of computing, the spare capacity has never been left unused. In addition, one should notice that the capacity and speed of wired connections has increased much faster. For each magnitude of improvement in wireless communications there has been an improvement of 2-3 magnitudes in wired communications.

The problems of wireless links are not uniform. We have wireless LANs, satellite links, cellular networks, and short-range radio links. Each poses specific problems of its own. Hence, the support system of a nomadic user must be able to support communication links of different kinds. It must enforce the higher layers of communications to adapt to the situation at hand. However, the adaptation of communication is not sufficient, the behavior of applications also needs to be adapted.

Distribution of Functionality

Situations, in which a user moves with her end-device and uses information services, are challenging. However, the nomadic user of tomorrow will not appreciate a static binding between her and an access device; not even in the case of multi-mode access devices that can handle several access technologies including wireless LAN, short-range radio, and packet radio. It must be possible to move a service session (or one end-point of a service session) from one device to another.

In these situations the partitioning of applications and the placement of different co-operating parts is a research challenge. The support system of a nomadic user must distribute, in an appropriate way, the parts among the end-user system, network elements and application servers. In addition, when the execution environment changes in an essential and persistent way, it may be beneficial to redistribute the co-operating parts.

The redistribution or relocation as such is technically quite straightforward but not trivial. On the contrary, the set of rules that the detection of essential and persistent changes is based on is a challenging research issue.

INTERNET PROTOCOLS AND MIDDLEWARE

Figure 2 decomposes the abstract notion of execution environment in Figure 1 into a layered architecture. It should be noticed that the operating system is not shown in Figure 2 because it would introduce another dimension. In most operating systems the layers below the middleware are encapsulated into program libraries that are available through system calls, file descriptors, etc.

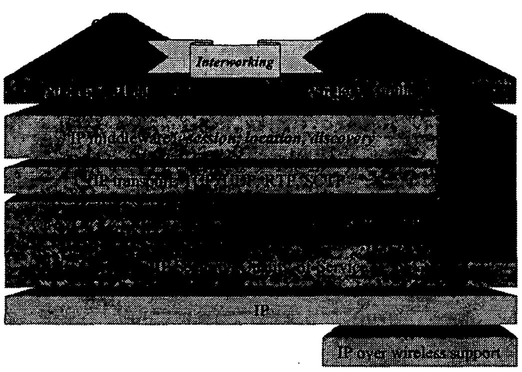

Figure 2: Layered Architecture of Middleware and Internet Protocols

The programming model of the Internet is based on sockets, protocol specific APIs and state machines of application protocols. The state machines need to be separately implemented for each protocol. Although the Internet approach is efficient for a single protocol, it is very cumbersome and expensive for application development that involves several operating systems and protocols. Moreover, many Internet protocols do not explicitly define the presentation format on the wire, which has introduced serious interoperability problems between implementations. Another concern is that the Internet approach is prone to feature interactions since many protocols are developed independently from others.

In middleware solutions, particularly in CORBA, a considerable amount of effort has been put on interoperability in heterogeneous operating environments. This has resulted in a uniform baseline protocol that also specifies the presentation format. In the case of CORBA, the unifying protocol is GIOP or Generic Inter-ORB Protocol and the presentation format is CDR or Common Data Representation that specifies how different data types and complex data structures are put on the wire. Therefore, many details that need to be handled by applications in the Internet solution are handled by the platform in the middleware solution.

It should be noted that in the Internet world there is also a "middleware" initiative called Internet2 Middleware [8]. However, its focus is on APIs for a set of core services including identifiers, authentication, directories, certificates, and authorization. Therefore, its programming model is quite primitive when compared to that of CORBA, Java, or DCOM.

When the CORBA approach—as well as the Java and DCOM approach—is compared to the Internet approach, the only possible conclusion is that the middleware solutions provide a much more convenient programming model than the Internet solution does. The middleware solutions are usually based on object-oriented programming and method invocations. The invocations are based on strongly typed interfaces that provide both compile and run time error checking. They also hide many implementation details. Therefore, middleware-based application development is much faster than the Internet based one.

Although the middleware solutions provide superior programming environments over the Internet solution, they also pose serious shortcomings. The fundamental problem of the current middleware specifications is that they only take advantage of a narrow subset of useful Internet protocols. The current middleware specifications were born in a time when the Internet protocols was a synonym of the TCP/IP transport. Later they have developed solutions of their own for Quality-of-Service, directory, discovery, and so on, independently from each other and from the IETF specifications.

RESEARCH ISSUES

From the software architecture point of view, the fundamental research challenge is the question of how the developments in Internet protocols and in different middleware solutions could be harmonized. This requires both an overall vision on a broad frontier of standardization activities and deep insight into various specifications. In particular, we do not want to discard existing middleware solutions since they are well-established and provide obvious benefits for application development.

By harmonization we mean two things. Firstly, we need to solve the problem of incorporating evolving Internet solutions of Quality-of-Service, mobility, discovery, and security into the existing middleware specifications without breaking those specifications. Secondly, we need to find solutions to how different middleware solutions can become interoperable in the sense that components of an application can be executed on different middleware platforms.

Developments in Internet Protocols

The problem space that the IETF addresses is very broad. Of the c. 120 active working groups in the IETF, the most important ones for middleware are depicted in Table 1. In the table the working groups are classified under labels of Quality-of-Service, Mobility, Discovery, Security, and Transport. The labels and groupings are more or less arbitrary but they reflect separate areas of competence needed in middleware development. The number of working groups mentioned in the table and the diversity of their scope call for co-ordination. When the other standardization bodies relevant to telecommunications are included, the co-ordination becomes really challenging.

Table 1: IETF Active Working Groups Closely Related to Middleware

Quality-of-Service
policy: Policy Framework
diffserv: Differentiated Services
issll: Integrated Services over Specific Link Layers
rap: Resource Allocation
mpls: Multiprotocol Label Switching

Mobility
mobileip: IP Routing for Wireless/Mobile Hosts
manet: Mobile Ad-hoc Networks
nasreq: Network Access Server Requirements
roamops: Roaming Operations
seamoby: Seamless Mobility

Discovery
dhc: Dynamic Host Configuration
srvloc: Service Location Protocol
zeroconf: Zero Configuration Networking
ldapext: Lightweight Directory Access Protocol (LDAP) Extension
ldup: LDAP Duplication/Replication/ Update Protocols
slop: Spatial Location Protocol

Security
ipsec: IP Security Protocol
ipsp: IP Security Policy
pkix: Public-Key Infrastructure (X.509)
spki: Simple Public Key Infrastructure
aaa: Authentication, Authorization and Accounting

Transport
ipngwg: IP Next Generation (IPNG)
ngtrans: Next Generation Transition
tsvwg: Transport Area Working Group
avt: Audio/Video Transport
pilc: Performance Implications of Link Characteristics
ecm: End-to-end Congestion Management
sigtran: Signaling Transport

In the IETF there are two questionable tendencies. The first one is the popularity of the NIH [9] or "not-invented-here" attitude, which leads to duplication of work and to neglect of existing and mature software solutions. The second one is the willingness to develop a new protocol for each new problem. The consequence is a huge number of protocols that need to be implemented, tested, and maintained. In the fixed network this is primarily only waste of manpower. On small terminal devices the problem is much more serious. We should not waste all available memory on the control and management planes since the user plane hosts the end-user applications that are the only thing that the end-users are willing to pay for.

Instead of separate, stand-alone protocols, we are advocating a base protocol approach. A good candidate for the base protocol would be the GIOP specified in the OMG. HTTP together with XML—through the W3C Simple Object Access Protocol or SOAP initiative [10], for example—might be an alternative to the GIOP but the problems of HTTP/XML over wireless links are much more serious than those of GIOP. The same is also true for Java RMI over wireless [11].

Interoperability Between Middleware Solutions

The diversity of the 3G and 4G devices—terminals, network elements, and application servers—imply that different middleware platforms are most appropriate for different devices and purposes. This heterogeneity requires interoperability on two levels: between middleware platforms and between parts of an application running on different middleware platforms.

The interoperability between different platforms is quite mature. In particular, the OMG has been the leading forum in specifying interoperability bridges between CORBA and other middleware platforms. In contrast, the interoperability between parts of an application running on different middleware platforms is still immature. There are practically no tools available to support this kind of interoperability. The burden of interoperability is totally left to application developers.

The OMG, however, has started a comprehensive development of a new architecture denoted as *Model Driven Architecture* (MDA) [12]. The objective is to interrelate IDL specifications, UML modeling [13], Meta-Object Facility [14], and XML Metadata Interchange (XMI) [15]. The forthcoming MDA might provide a useful starting point for tools supporting interoperability between parts of an application running on different middleware platforms.

Network Evolution

The UMTS Release 5 is launched as an all-IP network. Due to the shortage of permanent IPv4 addresses, embedded and optimized support for security, mobility and Quality-of-Service in IPv6, the IPv6 will be the basis of future mobile networks. However, this is not free of problems. Many existing Internet applications are tied to IPv4 addressing and identification. Therefore, transition to IPv6 has met some hesitation.

The IETF has taken the transition seriously. The Next Generation Transition (ngtran) working group [16] has produced several alternative transition paths. These include A SOCKS-based IPv6/IPv4 Gateway Mechanism, Dual Stack Transition Mechanism, An IPv6-to-IPv4 transport relay translator, IPv6 over IPv4 tunnels for home to Internet access. A good summary can be found in RFC 2893 [17].

The transition problem originates in the Internet programming model and in the unfortunate design decision to use the same entity as the routing address and the identity. In principle, the middleware should not have this problem since the transport mechanism is separated from the entity identity. Unfortunately this is not true in practice since most middleware solutions implicitly use the IP-address in entity identity. However, the coupling of routing address and entity identity is much weaker in the middleware solutions and, therefore, the transition problem is easier to solve.

Most of the middleware solutions are based on an entity reference of opaque nature. Therefore, if the middleware decouples the reference (i.e. entity identity) from the routing address, then the applications are portable from IPv4 networks to IPv6 networks. In OMG the work is going on. The RFP on GIOP SCTP/IPv6 protocol mapping [18] together with the RFP on Extensible Transport Framework for Real-Time CORBA [19] and the joint response [20] to the RFP on Wireless Access and Terminal Mobility in CORBA are steps towards a generic solution.

CONCLUSIONS

The main implication of all IP networks on the middleware is the increasing richness of functionality and features available in the IP networks. This implies that the middleware standardization bodies need to take another look at their solutions for Quality-of-Service, mobility, discovery, and security. They should also consider alternative Internet transport mechanisms to TCP/IP, which is a good protocol for bulk data transfer but not for messaging. When the middleware specifications are based on the IP solutions for Quality-of-Service, mobility, discovery, and security, interoperability and interworking will be much easier to arrange.

The main counter-implication is that the IETF should not be burdened with duplicating proved solutions. In any case, if the IETF continues with the NIH-attitude, then we should not adopt the IETF specifications that replace middleware specifications for which superior application programming models are available.

It is highly improbable that there will be—in a quite near future—a single dominant middleware platform. Therefore, the interoperability between middleware platforms and between parts of applications running on different middleware platforms is of crucial importance.

REFERENCES

1. Network Policy and Services: A Report of a Workshop on Middleware. RFC 2768, February 2000.
2. http://www.omg.org/
3. http://java.sun.com/j2ee/
4. http://java.sun.com/j2me/
5. Eddon, G. and Eddon, H. Inside Distributed Com (Microsoft Press, 1998).
6. http://www1.wapforum.org/tech/documents/WAP-195-WAEOverview-20000329-a.pdf
7. http://www.parlay.org/
8. http://www.internet2.edu/middleware/
9. Frakes, W.B. and Fox, C.J. Sixteen questions about software reuse. Communications of the ACM, Vol. 38, No. 6, pp. 75-85, June 1995.
10. http://www.w3.org/TR/SOAP/

11. Campadello, S., Helin, H., Koskimies, O., and Raatikainen, K. Wireless Java RMI. http://www.cs.Helsinki.FI/research/monads/papers/edoc2000/rmi.html
12. OMG discussion paper (version 3.2) on Model Driven Architecture, OMG document omg/00-11-05.
13. UML Specification. OMG document formal/2000-03-01.
14. MOF Specification. OMG document formal/2000-04-03.
15. XMI Specification. OMG document formal/2000-11-02.
16. IETF ngtrans Working Group. http://www.ietf.org/html.charters/ngtrans-chapter.html
17. IETF. Transition Mechanisms for IPv6 Hosts and Routers. RFC 2893.
18. OMG. GIOP SCTP protocol Mapping RFP, OMG document/2000-12-18.
19. OMG. Extensible Transport Framework for Real-Time CORBA The RFP is OMG document orbos/2000-09-12. The initial submissions are in OMG documents orbos/2001-01-05, orbos/20001-01-13, and orbos/2001-02-05.
20. Joint revised submission to the OMG RFP on Wireless Access and Terminal Mobility in CORBA. OMG document telecom/2001-02-01.