Large-Scale International IPv6 Pilot Network

IPv6 and e-business

Jérôme TARTE
IBM Systems & Technology Group, Client Technology Center for Lab Services
EMEA ATS e-Business Solutions Center (eBSC), La Gaude, France
Jerome.tarte@fr.ibm.com

Executive summary

After a quarter century of use, today the IPv4 protocol is showing its limitations. With the explosion of Internet and the growth of IP address needs, a new version of the IP protocol is needed. This new version of the protocol, IPv6, adds some features to handle the limitations of IPv4, like QoS, plug and play support, improved support for mobile clients and the multiplication of IP addresses.

Java, the main language for e-business applications, is ready for the support of IPv6. The last releases of the JDK provide APIs compliant with the IPv6 protocol. IBM, pioneer of e-business and On Demand Business trends, is progressively enabling the IPv6 support into its software now that is supported in the operating systems of its servers. The first IBM software that is fully enabled with IPv6 support is WebSphere Application Server v6.

The application server is the main piece of an e-business infrastructure. It provides the execution environment for the e-business applications. It handles the network connectivity. Running in an application server, the e-business applications are not directly dependent on the IP protocol version. Thus, today, the e-business applications are ready for IPv6 as soon as their underlying middleware such as Java is ready. The migration of e-business applications from IPv4 to IPv6 infrastructure can be achieved without major work. They can also work in mixed infrastructure (IPv4 and IPv6). This is an important point because the deployment of IPv6 will be progressive and the two versions of IP must live together.

Table of Contents:

I. Introduction

IPv6 is the new version of the network protocol. It is here because IPv4, the most used version of the protocol, has shown its limitations. With the introduction of e-business On Demand, IT infrastructure must produce services that offer more flexibility, performance and security. The network, as part of the infrastructure, will be an active participant in this transformation. IPv6, with its new features, contributes to the IT transformation initiated by e-business On Demand.

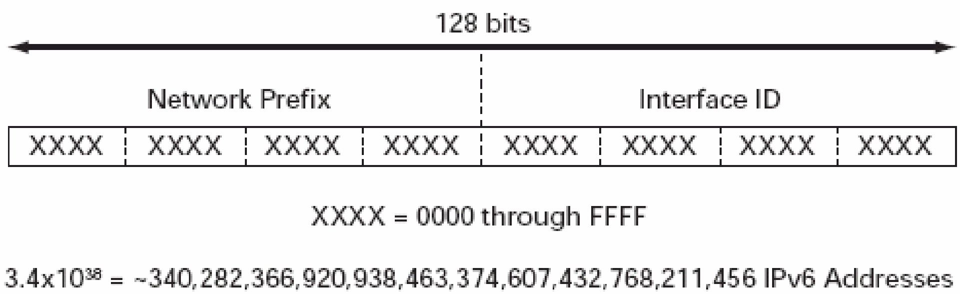
IPv6 is a network protocol. But in an IT System, everything is not network-centric. Many application developments are designed to use the network regardless of how the header of IP packets is organized. The development team uses APIs that mask the low level network-specific details.

This document studies IPv6 features from the viewpoint of e-Business competencies and the needs of On Demand Business, rather than from that of a network specialist. After listing the interesting features of IPv6, this document describes the support of IPv6 in Java, the preferred language of e-Business. It will give also a view of the IPv6 support provided by IBM. At the end, it explains how e-business applications support the IPv6 protocol.

II. IPv6 features

Expanded routing and addressing

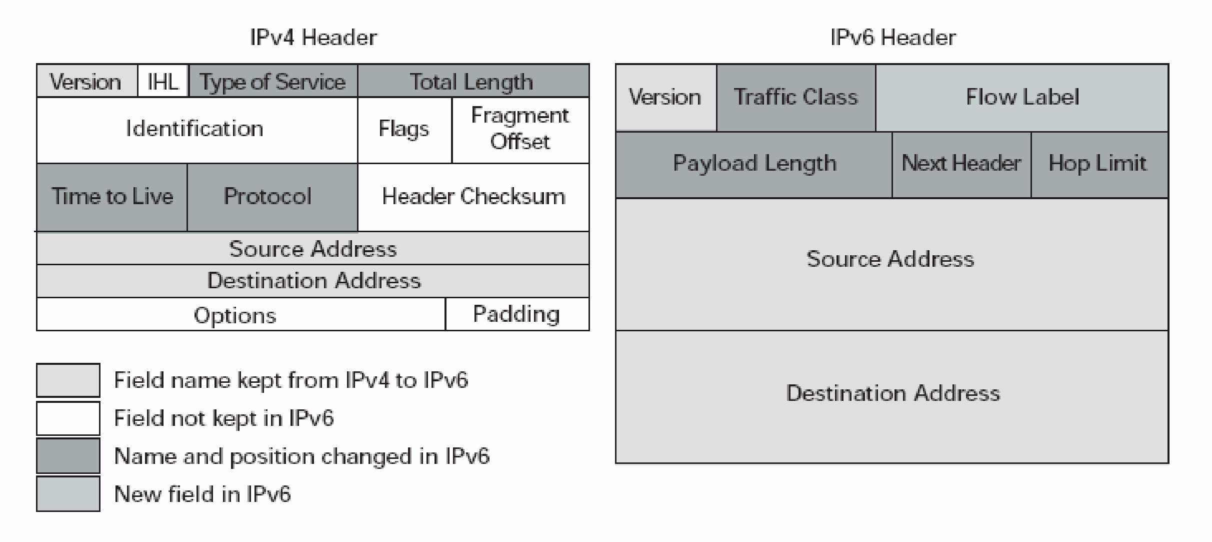
Currently, the major problem of IPv4 is the limited number of IP addresses. The offer is below the demand. To avoid this problem, IPv6 increases the number of address bits by a factor 4, from 32 to 128 bits. Coding the address in 128 bits theoretically gives approximately 3.4×10^{38} addressable nodes. On this planet, this situation theoretically allocates about 10^{28} address per person. Of course, in reality the address has some structure as shown below and the number of addresses available in practice is vastly less, but still largely adequate.



The profusion of addresses allows each node (server, mobile device, network equipment …) to have a unique address. The network infrastructure with network address translation (NAT) becomes redundant. (Note that the security advantages sometimes claimed for NAT can easily be met using safe router configurations with IPv6.)

Header Format Simplification

In terms of bits, the header of an IPv6 packet is longer than that of an IPv4 packet, but if you compare its structure, it is simpler.



The IPv6 header has a fixed size of 40 octets. If you compare with the basic IPv4 header size, it is twice as long. But the increase is due to the increased size of address (from 32 bits to 128). In IPv6 some header fields have been removed, like Header Length and Fragment Offset. This removal results in faster processing of the basic IPv6 header, and routing efficiency is the same as or better than IPv4. Overall performance may also depend on the optional headers in use. In general, the simpler header is also an advantage for efficient compression on wireless links. Finally, the fields in the IPv6 header are conveniently handled in 64 bit segments, taking advantage of the current generation of 64-bit processors.

Plug-and-Play support

Address auto-configuration is built into the IPv6 protocol to facilitate intranet-wide address management, enabling a large number of IP hosts to easily discover the network and get a new and globally unique IPv6 address associated with their sub-net location. The auto-configuration feature enables “plug-and-play” Internet deployment of new consumer devices, such as cell phones, wireless devices, home appliances, and so on. As a result, network devices could connect to the network without manual configuration and without any configuration servers, such as DHCP servers. (DHCP is also available for IPv6, if preferred.)

A router on the local link will send network-type information, such as the prefix of the local link and the default route in its router advertisements. The router provides this information to all the nodes on the local link. As a result, a host can auto-configure itself by appending its 48-bit link layer address (MAC address) in an extended universal identifier EUI-64-bit format to the 64 bits of the local link prefix advertised by the router.

Authentication and privacy capabilities

IPSec is an optional part of IPv4 but it is mandatory in IPv6 and is part of the IPv6 protocol suite. IPSec can be enabled in each node of IPv6. This feature increases the security of the network.

By providing global unique addresses, avoiding the ambiguous address needed in IPv4, IPv6 can assure end-to-end security services by providing access control, confidentiality and data integrity mechanisms. This mechanism makes the usage of a firewall not mandatory in all situations and avoids the additional problems added by this equipment like performance bottlenecks. IPv6 provides security extension headers, making it easy to implement encryption, authentication, and virtual private networks (VPNs). (Nevertheless, it is likely that most enterprises will prefer to retain a firewall for routine traffic, at least in the early years of IPv6 deployment.)

Quality of service (QoS)

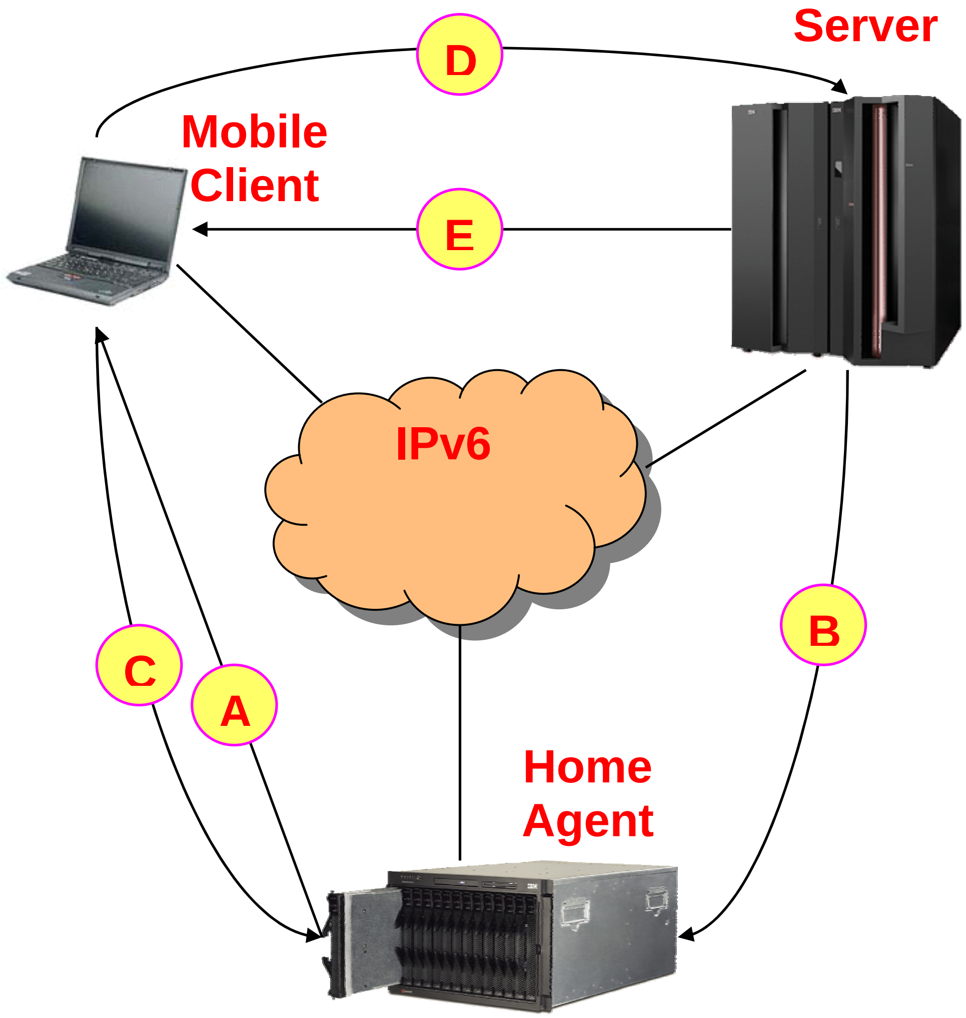
The QoS support in IPv4 is continued in IPv6. The support for class of service is still available with the Traffic Class field in the IPv6 header. However, the IPv6 header introduces a new field: the Flow Label. This field is used to contain a label identifying a specific flow. The label is generated by the source node and allows downstream nodes to take appropriate action. At the time of writing, QoS support based on the Flow Label remains to be completely defined.

Improved Support for mobile clients

All IPv6 nodes can potentially implement mobility as needed because the standards for Mobile IPv6 have been defined by the IETF. Mobile IP defines how the mobile device can move around without breaking its connection or changing its IPv6 address.

All hosts will include support for communicating directly with mobile clients, without having to send packets through an intermediate proxy. IPv6 packets addressed to the home address of a mobile node are transparently routed to its care-of address through the caching of the binding of its home address with its care-of address. This binding allows any packets destined for the mobile node to be directed to it at this care-of address.

The routing headers in IPv6 make Mobile IPv6 much more efficient for end devices than Mobile IPv4. The use of the routing header for Mobile IP, rather than IP encapsulation, enables Mobile IP to avoid triangle routing, making it much more efficient in IPv6 than in IPv4.



Operations sequence for mobility:

1. As the mobile client roams, it notifies its Home Agent of its current location by sending its Care of Address (A).
2. When a server wishes to communicate with the mobile client and does not already know the current Care of Address, it sends a packet to the mobile client's home address (B)
3. The Home Agent intercepts the packet and forwards it to the mobile client at its current Care of address (C)
4. The mobile client sends a response directly to the server and includes its Care of Address (D)
5. The client and the server send packets directly to one to another without having to send packets through the Home Agent (D), (E).

III. IPv6 and Java

At the time of writing, we have the possibility to work with two levels of Java, the official 1.4 and the next generation 1.5, provided as beta. These two JDKs allow working with IPv6. The level of IP v6 implementation in the JDK 1.5 is better. This chapter describes the functionalities of these JDK's IP v6 features.

JDK 1.4

With the J2SDK/JRE 1.4 release, IPv6 support has been added to Java Networking. This will make J2SE compliant with the following specifications (RFCs):

- RFC 2373: IPv6 Addressing Architecture;
- RFC 2553: BasicSocket Interface Extensions for IPv6;
- RFC 2732: Format for Literal IPv6 Addresses in URLs.

Since the J2SDK does not support raw sockets, RFC 2292 (: Advanced Sockets API for IPv6) is not supported in this release.

Other interesting features of IPv6, such as tunneling, auto configuration of addresses, mobile IP, etc., are not supported at the Java API level, as they are handled automatically by the underlying OS or system support.

On systems with a dual stack, system properties are provided for selecting the preferred IP stack. By default, the IPv6 stack is preferred because the IPv6 Socket can work with IPv4 and IPv6 peers on a dual stack system.

All the methods for programming TCP/IP applications are localized in the java.net package. In this package, the class *InetAdress* has been modified to support both IPv4 and IPv6 addresses. Two new classes appears, *Inet4Address* and *Inet6Address*, each class inherits from *InetAdress* and implements the specific behavior of its protocol version. As Java is an object oriented language, an application need to deals with the *InetAddress* class. With the polymorphism, it will get the correct behavior. New methods are introduced to be able to test the nature of IPv6 address (Site, Org address…).

Due to object oriented nature of Java, the socket classes work both IPv4 and IPv6 addresses. In fact, the classes manipulate an *InetAddress*. The socket API can handle IPv4 and IPv6 traffic. The selection of IP stack depends to OS where the application run and the user's stack preference setting. To provide the same support on IPv4 as IPv6, the old socket API has been overloaded to support the two protocols options.

JDK 1.5

The JDK 1.5 increases the support of IPv6. It keeps all the functionalities introduced with the JDK 1.4 and provides some new features.

Arguably, the major enhancement resides in the introduction in the socket API of the method *setPerformancePreference*. This method allows the application to express its own preferences to tune the performance characteristics of this socket. Performance preferences are described by three integers whose values indicate the relative importance of short connection time, low latency, and high bandwidth. With this method, the network oriented notion of Quality of Service (QoS) is introduced. Any application can set its preferences to adapt its network traffic and provide the best QoS.

IV. IPv6 and IBM

IBM's contribution to IPv6

As a participant in the IETF and IPv6 forum, IBM is an active actor in the design and standardization of IPv6 since 1995. All the operating systems provided by IBM support IPv6. AIX was the first one in 1997. By 2002, the support was extended to its zSeries, pSeries and iSeries operating systems. Concerning Linux, IBM Linux Technology Center has contributed in the support of IPv6 in the Open Source operating system, especially in the area of mobile IPv6

IBM was an active participant in the European Union's 6NET project. With 35 other industrial and research partners, IBM participated in the promotion and the deployment of IPv6. IBM led the work on middleware and applications trials, which included the use of both IBM WebSphere and the Open Grid Services Architecture. Several entities of IBM, such as IBM Global Services, IBM Systems and Technology Group and IBM Software, were all involved in this project.

IBM is working to extend the support of IPv6 to cover all relevant software products. The requirements for IPv6 support have been identified. Some key products (IBM WebSphere Application Server v6) have already been enabled. IBM is continuing to enable its other products (the next version of IBM Tivoli Access Manager is pre-announced with IPv6 support). The timing of product enhancements will depend on individual product release schedules and marketplace needs. IBM will continue to build on these foundations to ensure that the benefits of IPv6 connectivity become available to all its customers.

IBM WebSphere Application Server V6

Internet Protocol Version 4 is no longer viable for many businesses, especially in emerging economies with limited IPv4 infrastructure. Because it is based on a 32-bit architecture, there is a growing shortage of Internet Protocol Version 4 (IPv4) addresses. Internet Protocol Version 6 (IPv6) is based on a 128-bit architecture, which allows a far greater number of addresses to be available for use over the Internet. In response, WebSphere Application Server Version 6 now includes support for IPv6, in addition to continued support for IPv4. This means that nodes running WepSphere Application Server Version 6 can use IPv6. (Nodes running WebSphere Application Server Version 5.x cannot use IPv6.)

WebSphere Application Server Version 6 supports a *dual mode* environment in which you can have older legacy applications running on IPv4 and IPv6-enabled applications running on IPv6. However, there are restrictions on using IPv4 and IPv6 in the same cell. This chapter documents those restrictions as well as outlines the ways in which you can set up your cells, depending on the version of IP that you will be using.

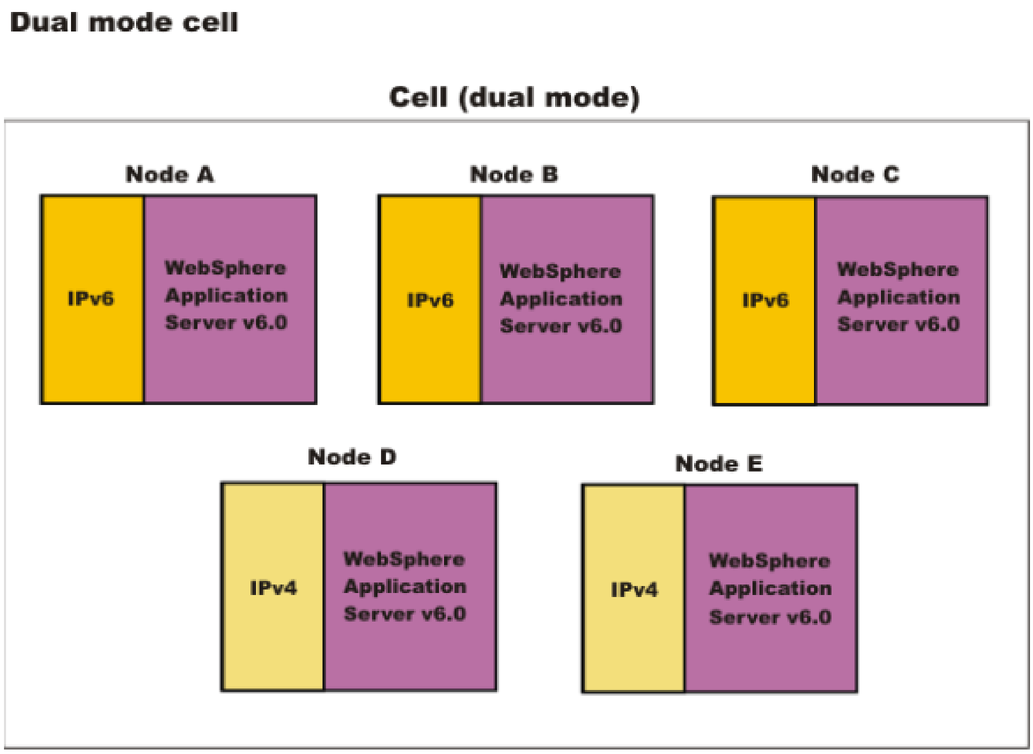
From an IP perspective, you must adhere to one of the following scenarios:

Dual mode cell

In a dual mode cell, mixed IPv4 and IPv6 communications are supported. By default, a cell is set to dual mode when it is created. But note that only nodes running WebSphere Application Server Version 6 are valid in a dual mode cell.

IPv4 and IPv6 nodes cannot communicate directly with each other, so the purpose of the dual mode cell is to enable this communication, thereby allowing you to use your existing applications, running over IPv4, with newer applications that have been enabled for IPv6.

The following illustration shows a dual mode cell:

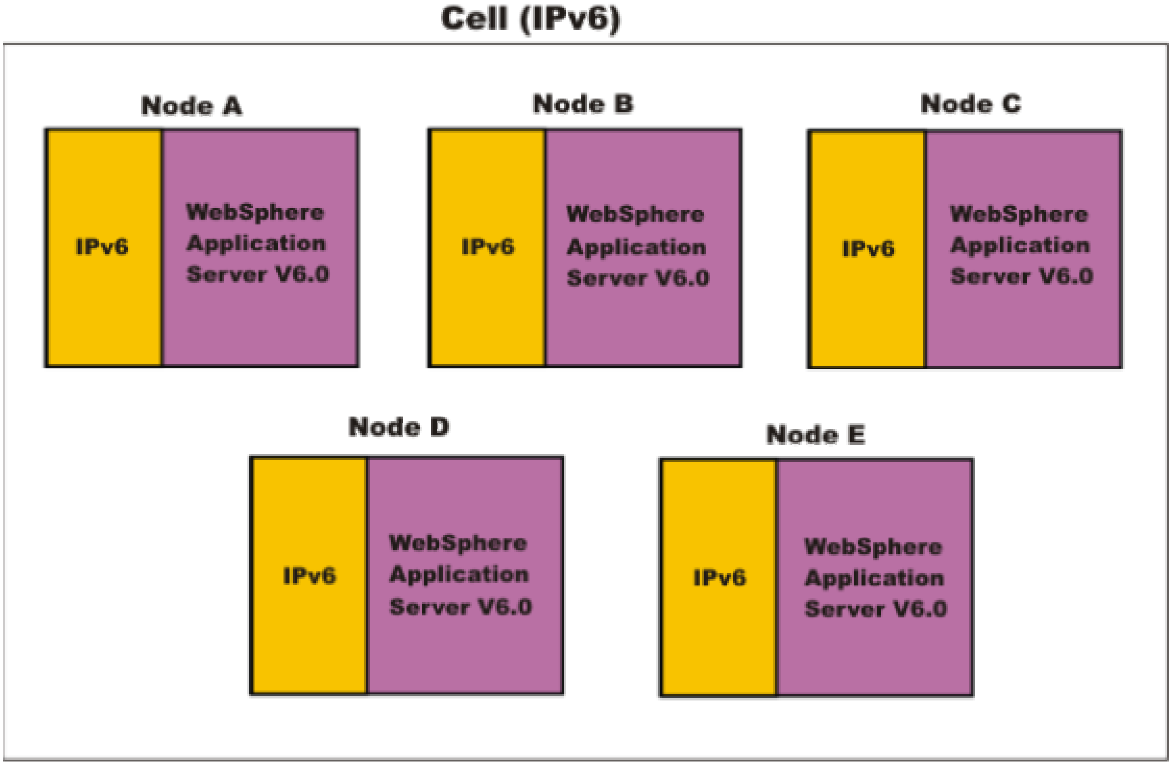


IPv6-only cell

In an IPv6-only cell, all nodes must:

- Use IPv6
- Run WebSphere Application Server Version 6
- Have host names defined as strings or 128-bit numerical addresses.

IPv6-only cell



V. IPv6 and e-business

What is e-business?

One of IBM's activities on the 6NET project was to provide e-business applications to show how this kind of application can use IPv6 features and support On Demand Business. But before studying this point, it is necessary to define what is covered by the term "e-business".

Introduced at the end of the 1990s by IBM, e-business is a concept that represents the execution of real-time business processes with the assistance of Internet technologies. An e-business application can be an application that offers the possibility to manage a bank account with an Internet Browser or an application selling goods or services without the intervention of a salesperson. In terms of technology, e-business is usually associated with Java and an application server. By over-simplification, e-business is often a synonym for Java applications running on an application server infrastructure.

Today, IBM proposes an evolution of its e-business strategy: e-business on demand. In the emerging environment of e-business on demand, where information technology becomes a resource supplied by the network when the user wants it, the traditional boundaries between enterprises, service providers, and end users melt away. IBM defines an on demand business as an enterprise whose business processes, integrated end-to-end across the company and with key partners, suppliers and customers, can respond with speed to any customer demand, market opportunity or external threat. In this networked environment, any user may need to access any service of any service provider.

Challenges in linking e-business to IPv6

The main challenge is to link network features to business applications. The main question is how to translate low level network features (OSI Model level 3) into a specific impact on the high level (level 7 application).

An e-business application is developed by respecting standards like J2EE and Web Services.. If you respect these standards, an application may be developed to run on any application server supporting those standards. The application server, such as IBM WebSphere Application Server, has the responsibility of masking the environment from the application.

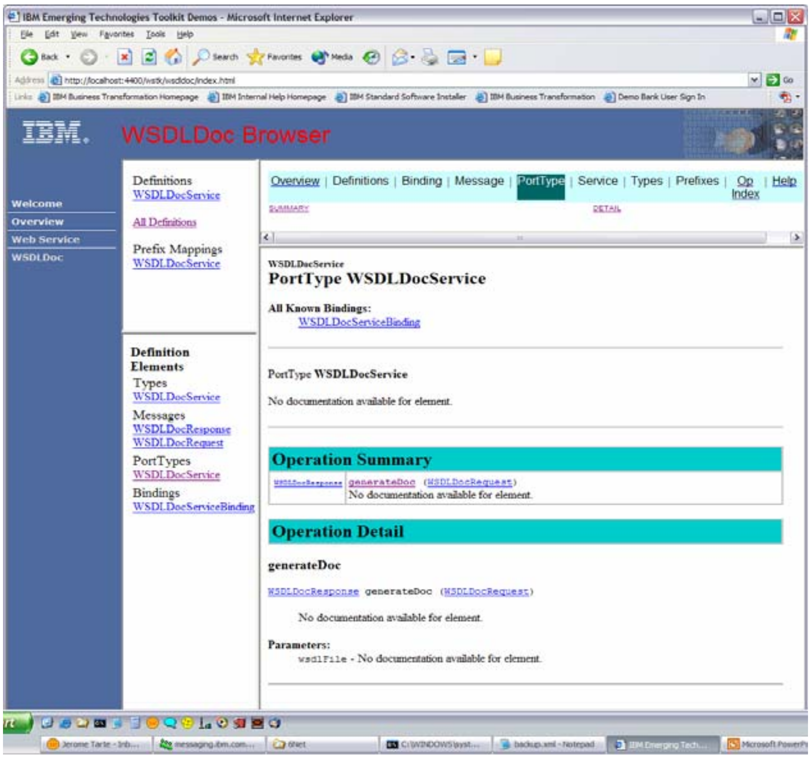
E-business applications on 6NET project

To handle the previous point, the IBM activity in 6Net was not mainly focussed on the development of new e-business applications but on the migration of existing applications to an IPv6 environment. This choice was motivated by several reasons. The first one was the difficulty in developing an e-business application using specific IPv6 features. An e-business application runs on an application server that manages the connectivity and network aspects and hides the IP layer. The second reason resides in the fact that the migration of networks from IPv4 to IPv6 infrastructures will not be instantaneous. During a long period, the two versions of IP protocol will have to coexist. Today’s e-business applications are running on IPv4 infrastructure. Tomorrow most of them will have to run on IP v6 infrastructure. The migration must require minimal effort and not interfere with simultaneous use of IPv4 and IPv6.

The lessons learned during the 6NET project have been collected during the migration of several e-business applications from an IPv4 environment to an IPv6 environment. All of these applications were developed and tested first in an IPv4 environment. Once they were validated in this first environment, they were then deployed on IPv6 infrastructure. The following paragraphs give a short description of these applications.

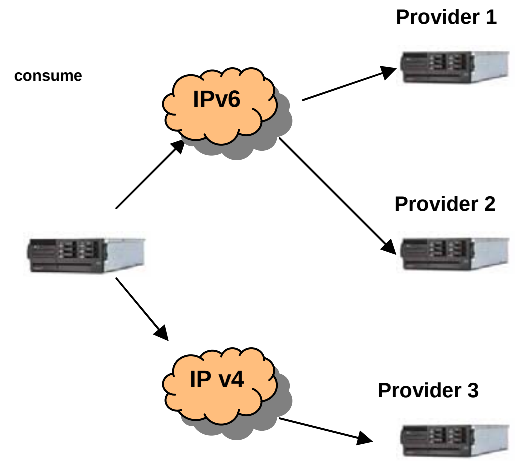
Web Service for the automatic documentation of Web Services

This demonstration shows the generation of HTML documentation from a WSDL file. There are multiple interfaces to this tool: Web interface, Web Services. The service can use multiple URL formats (IPv4 URL and IPv6 URL) to localize a WSDL file.



Buyer-Seller: end-to-end purchase of electronic components

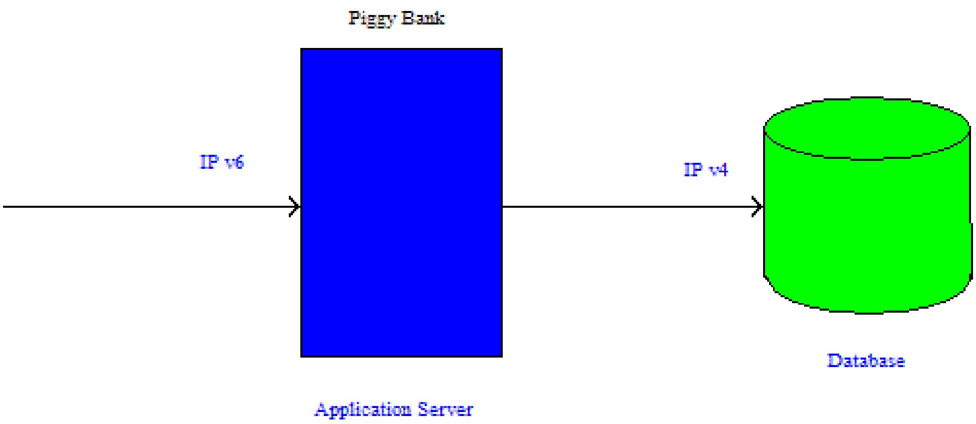
This Web Services demonstration shows how it is possible to locate potential suppliers, agree on a product part number, obtain the lowest price from several sources, place an order, etc. Communication between the consumer and the providers is made through Web Services. It demonstrates the capability for a consumer to work with IPv6 and IPv4 Web Services.

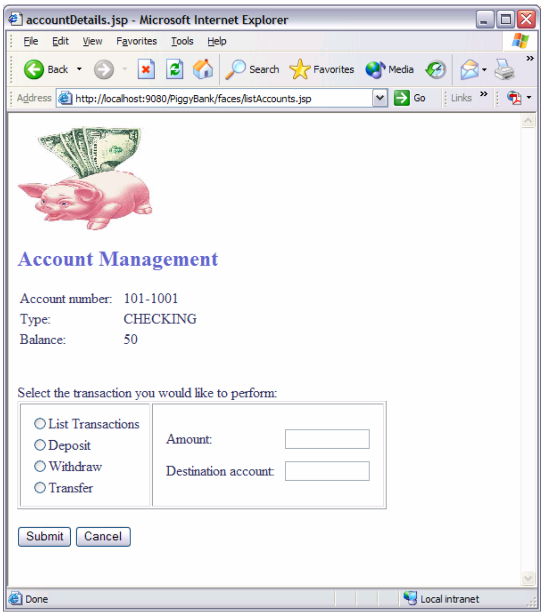
Piggy Bank: bank account management

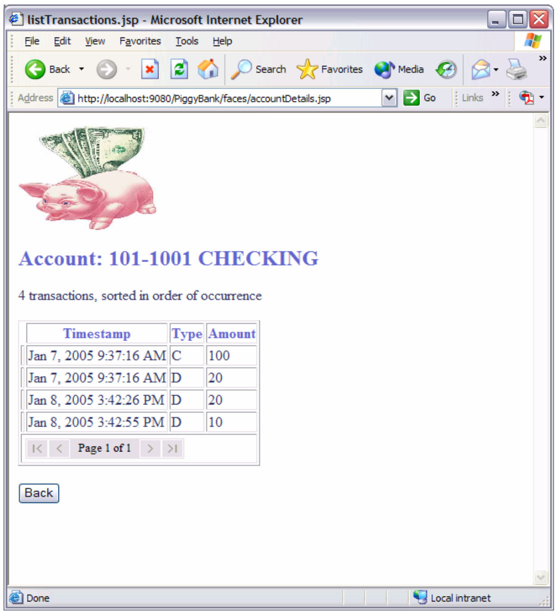
This demonstration shows how it is possible to view an account and to do a transfer, deposit and withdrawal using a J2EE application. The account is stored in DB2, IBM's SQL-compatible database.

It is a demonstration of an IPv6 application communicating with a database running IPv4.



The user interface is built with Java Server Face (JSF) pages, the latest technology for producing thin web user interfaces.

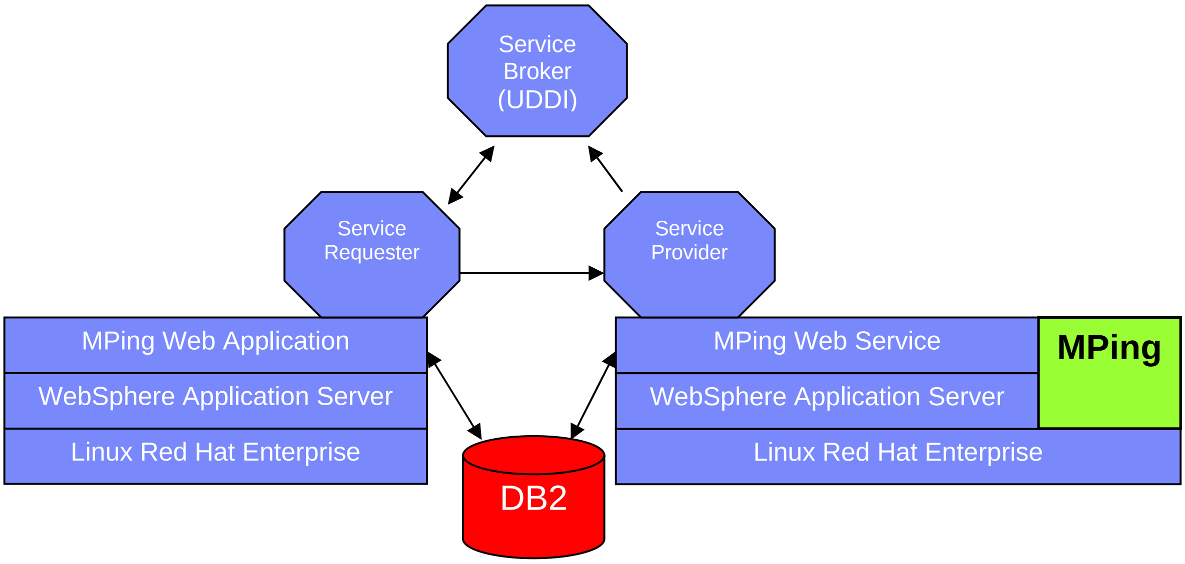




Multi-Ping (Mping) Web Service

Written by UNINETT, the MPing tool includes a system for collecting statistics about packet delay and loss in a TCP/IP network using the Internet Control Message Protocol (ICMP) "Echo" facility to measure round-trip delays and packet loss across network paths. IBM has developed and published a Web Service that wraps Mping components “on demand”. This Web Service allows everyone to invoke an Mping session without installing the software on his or her own machine.

The structure of the service is shown below.



The MPing Web Application and MPing Web Service have been installed on two different servers to demonstrate the capability to access to the Web Services from a remote location. The two Web Applications work with a database that is accessed by JDBC over IPv4. The MPing Web Service writes the result of ping statistics. The MPing Web Application read the database to present the results of Mping session.

VI. Lessons learned

The first result of the IBM activity on 6NET is the confirmation of the middleware role. The Application Server, such as IBM WebSphere Application Server, provides an execution environment for e-business applications. This execution environment has the responsibility to handle some basic functions like network connectivity. By handling these functions, the middleware provides an environment where the application can be written independently of these aspects. An e-business application can be developed without specific consideration of the network infrastructure. The middleware is in charge of translating its network needs to the available IP protocol.

The second result is a consequence of the first. The existing e-business applications that have been developed in an IPv4 context can be migrated without actions from the development team in the great majority of cases. Only one case has shown a minor problem. The application Buyer-Seller has client-server architecture using an applet. The migration of this application needed an adaptation of the applet source code of the applet to manage IPv6 connectivity. But the problem was not really a problem of IPv6 support; it was more a design problem. With a correct design that enables the support of multiple IP protocols, IPv4 and IPv6, the migration is straightforward.

The major lesson learned is the possibility for an application to work well with a mixed IP architecture. This is an important point because, today, not all software components, like the database, support the IPv6 protocol. It is possible to construct a mixed infrastructure where some nodes are accessed in IPv6 and others in IPv4. By constructing this kind of mixed infrastructure, one can profit from IPv6 in the infrastructure without waiting for all the components to be IPv6-ready. And even this mixed mode does not require the middleware or application to be restructured. This is the advantage of constructing middleware using suitable abstractions (especially, only considering DNS names and never IP addresses) to hide network details. This possibility enormously simplifies transition planning, because not only is there no need for a "flag day" on which all network infrastructure changes simultaneously, but also there is no need for an application "flag day" either.

Today, e-business applications are not limited by the network infrastructure. With e-business on demand, the collaboration between different enterprises is enhanced. Any important e-business application will have interactions with several external components. With the progressive deployment of IPv6, these components may be accessed via IPv6 or IPv4. The choice of the protocol will depend on the choices made by various service providers in term of IT Infrastructure. The e-business applications can interact with the same service provided by two different service providers on two different IP protocol versions. The code of the e-business application requires no specific modifications to handle the two protocols.

VII. Resources

- IPv6 Forum : http://www.ipv6forum.org
- 6NET project : http://www.6net.org
- IBM IPv6 resources : http://www.software.ibm.com/ipv6
- IPv6 technical information : http://playground .sun.com/pub/ipng/html/ipng-main.html
- [JDK14v6] Java Development Kit 1.4.1, IPv6 Guide, http://java.sun.com/j2se/1.4.1/docs/guide/net/ipv6_guide
- [JDK15v6] Java Development Kit 1.5.0, IPv6 Guide, http://java.sun.com/j2se/1.5.0/docs/guide/net/ipv6_guide
- [RFC3493] R. Gilligan, S. Thompson, J. Bound, J. McCann, W. Stevens, Basic Socket Interface Extensions for IPv6, IETF RFC (obsoletes RFC2553), February 2003.
- [RFC3542] W. Stevens, M. Thomas, E. Nordmark, T Jinmei, Advanced Socket Application Program Interface (API) for IPv6 IETF RFC3542 (obsoletes RFC2292), May 2003
- [RFC2732] R. Hinden, B. Carpenter, L. Masinter, Format for Literal IPv6 Addresses in URL's, December 1999.
- [RFC2373] R.Hinden, S. Deering, IPv6 Addressing Architecture, July 1998
- Cisco Statement of Direction for IPv6 **http://www.cisco.com/warp/public/732/tech/ipv6/**