1 Preface

The *open*Net Server package includes the BS2000/OSD transport system. In addition to the proprietary NEA protocols, the BS2000 Communication Manager BCAM also supports ISO and TCP/IP protocols. While to date only Version 4 TCP/IP protocols have been supported, *open*Net Server V2.0 introduces the new Version 6 (IPv6) TCP/IP protocols in stage one, as described in this manual.

1.1 Target group

This manual is intended for anyone who

– has to decide on introducing IPv6 in BS2000/OSD,

– will be using IPv6 functionality on BS2000/OSD mainframes or

– wants to install IPv6 in BS2000/OSD.

Knowledge of the BS2000/OSD operating system as well as the basic TCP/IP concepts is therefore assumed.

1.2 Summary of contents

The introductory part of this manual provides a general description of the IPv6 functionality, which has been planned and agreed by the relevant bodies and then presents the BS2000/OSD-specific implementation of IPv6 stage 1 in the chapter on transition.

- Chapter 2 contains the most important facts about the development of the internet to date.
- Chapter 3 is intended for decision makers and provides an overview of the commercial requirements, the protocol fundamentals and the current status of IPv6 development as well as of the current debates for and against IPv6.
- Chapter 4 describes the functional and technical aspects of IPv6 in sufficient detail to be of interest to network specialists and programmers. With its detailed description, this chapter is also of interest to technically minded lay persons.

- Chapter 5 outlines the basic procedures used for converting from IPv4 to IPv6 and explains the transition using selected application scenarios.
- Chapter 6 describes the current status of implementation of IPv6 in BS2000/OSD. This functionality, which is implemented in IPv6 stage 1 is introduced from the perspective of generation, changes compared to IPv4 and usage and is illustrated with a number of examples.
- The appendix provides in-depth information on the topics of IPv6 address assignment and DNS usage.

2 Development of the internet

The internet as a set of interconnected and intercommunicating networks around the globe is a unique success story. Huge volumes of data are moved about daily over the internet by millions of users. Although the internet protocol family covers a variety of protocols, they are described generally by the term TCP/IP (Transmission Control Protocol and Internet Protocol). TCP/IP implementations are available on practically all operating systems and hardware platforms generally in use.

The origins of TCP/IP go back to the year 1968. The Advanced Research Projects Agency (ARPA) of the American Department of Defense (DoD) developed the ARPAnet at this time to enable shared use of resources for various members who were involved in different research projects. ARPAnet began operating in 1969. During the course of the next years, various weaknesses and inadequacies came to light in practical operation. The definition of the internet protocol was then defined in 1974 based on the experiences gained. Following various trial phases, TCP/IP was defined by the DoD in 1978 as the standard protocol for its data communication networks.

The advantages of IP technology are sufficiently familiar and undisputed. Internet technology provides an open standard for reliably connecting heterogeneous environments while remaining highly cost effective at the same time. Various services are involved here of which the World Wide Web and e-mail are undoubtedly the most important and have helped make the internet a consistent success.

However, the explosive growth of the internet did lead to problems in one particular area, and that was the assignment of internet addresses. The demand for IP addresses is growing enormously because of the increasing requests by companies and service providers for connections. All possible products will be internet-ready in the future. The manufacturers of such devices rely on having a sufficiently large address space. Theoretically, the 32 bit addressing facility of the IPv4 internet protocol offers the possibility of support for up to 4.3 billion terminals. However, the previous assignment of address space was not very efficient. Even at the beginning of the nineties it was clear that a solution would have to be found. Apart from expanding the address space, the research efforts concentrated on performance, simplified administration and routing as well as security aspects.

The result of this work was the specification of the new internet protocol alias IPng (Next Generation) alias IPv6 (the official name). IPv6 brings a series of advances: Apart from the considerable increase in address space from 32-bit addressing to 128-bit addressing, a variety of other important enhancements are taking place. These extend from integrated security functions to additional flexibility to plug-and-play functionalities and support for

realtime applications. All leading manufacturers of network technology have indicated their approval for the new IPv6 standard, hence clearing all doubts and paving the way for IPv6 as the standard of the future.

Already at the beginning of 1998, the worldwide IPv6 test network, the 6Bone, had reached 400 computers in 40 countries. There are over 50 different IPv6 implementations that are either already complete or are in development. Of these, some 25 different implementations are already in the 6Bone and in initial productive networks.

3 The business case for IPv6

Given the remarkable growth of the internet and the business opportunity it represents, IPv6 is of major interest for business development, from which both network providers and users will benefit.

3.1 IPv6 standardization and production status

IPv6 has been approved as a Draft Standard, which means that it is known to be highly stable and appropriate for productive use. A large number of end-users, standards groups, and network vendors have been working together on the specification and testing of IPv6.

Current Draft Standards include:

RFC 2373	IP Version 6 Addressing Architecture
RFC 2374	An IPv6 Aggregatable Global Unicast Address Format
RFC 2460	Internet Protocol, Version 6 (IPv6) Specification
RFC 2461	Neighbor Discovery for IP Version 6 (IPv6)
RFC 2462	IPv6 Stateless Address Autoconfiguration
RFC 2463	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification

Current Proposed Standards include:

RFC 1886	DNS Extensions to support IP Version 6
RFC 1887	An Architecture for IPv6 Unicast Address Allocation
RFC 1981	Path MTU Discovery for IP Version 6
RFC 2023	IP Version 6 over PPP
RFC 2080	RIPng for IPv6
RFC 2452	IP Version 6 Management Information Base for the Transmission Control Protocol

RFC 2454	IP Version 6 Management Information Base for the User Datagram Protocol
RFC 2464	Transmission of IPv6 Packets over Ethernet Networks
RFC 2465	Management Information Base for IP Version 6: Textual Conventions and General Group
RFC 2466	Management Information Base for IP Version 6: ICMPv6 Group
RFC 2467	Transmission of IPv6 Packets over FDDI Networks
RFC 2470	Transmission of IPv6 Packets over Token Ring Networks
RFC 2472	IP Version 6 over PPP
RFC 2473	Generic Packet Tunneling in IPv6 Specification
RFC 2507	IP-Header Compression
RFC 2526	Reserved IPv6 Subnet Anycast Addresses
RFC 2529	Transmission of IPv6 over IPv4 Domains without Explicit Tunnels
RFC 2545	Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
RFC 2590	Transmission of IPv6 Packets over Frame Relay
RFC 2675	IPv6 Jumbograms
RFC 2710	Multicast Listener Discovery (MLD) for IPv6
RFC 2711	IPv6 Router Alert Option

In addition, there are a number of other standards that are affected by IPv6 modifications and for which a revised version was therefore needed.

Examples of these include:

RFC 1888	OSI NSAPs and IPv6
RFC 2292	Advanced Sockets API for IPv6
RFC 2375	IPv6 Multicast Address Assignments
RFC 2450	Proposed TLA and NLA Assignment Rules
RFC 2471	IPv6 Testing Address Allocation
RFC 2553	Basic Socket Interface Extensions for IPv6

3.2 IPv6 design goals

Compared with IPv4, IPv6 offers a number of improvements, such as an extended address space and a simplified package layout.

The building, operation and updating of present day networks is not only resource intensive, it also requires enormous technical input. A great deal of attention has therefore been paid to creating autoconfiguration protocols for IPv6, minimizing the need for human intervention when assigning IP addresses and other network parameters.

A further benefit relates to the fresh start that IPv6 provides in the assignment of addresses, particularly in large, historically evolved networks. For instance, the routing hierarchy can be structured much more efficiently.

The much publicized attacks by hackers on networks, even of well-known software companies, demonstrate the need for improved security in IPv4-based networks. IPv6 offers improved authentication and encryption mechanisms, which take account of the increased need for security in the internet.

The following sections give an overview of the improvements that IPv6 brings to enterprise networking and the global internet.

3.2.1 Addressing and routing

IPv6 helps to solve a number of problems that currently exist within and between enterprises. For example, IPv6 will allow internet backbone designers to create a flexible and expandable global routing hierarchy. The functionality of the internet backbone, where major enterprises and Internet Service Provider (ISP) networks come together, depends largely on the ease maintenance of a hierarchical address system, similar to that of the national and international telephone systems. Large central-office phone switches, for instance, only need a national area code prefix to route a long-distance telephone call toward the correct local exchange.

Without an address hierarchy, backbone routers would be forced to store route table information on the accessibility of every network in the world. Given the current number of IP subnets in the world and the rapid growth of the internet, it is almost impossible to image how route tables and updates could be managed for so many routes. With an address hierarchy, backbone routers can use address prefixes to forward data and require very little routing information to do this.

Routing hierarchy in IPv4

In recent years, IPv4 has begun using the new routing technique called Classless Inter Domain Routing (CIDR) for implementing a routing hierarchy. CIDR permits route aggregation at various levels of the internet address hierarchy, whereby routers can store a single route table entry that provides accessibility to many subnets.

But CIDR does not guarantee an efficient and scalable routing hierarchy. In order to avoid maintaining a separate entry for each route, it is important for routers at lower levels of the backbone hierarchy to be collected together (or "summarized") into fewer and less specific routes at higher levels of the routing hierarchy. Legacy IPv4 address assignments that originated before CIDR and the current ISP hierarchy often do not allow summarization of routes. The lack of uniformity of addresses in the current hierarchical system, coupled with the rationing of IPv4 addresses, makes internet addressing and routing unnecessarily complicated. Furthermore, reassigning IPv4 addresses when changing from one ISP to another is complicated and thus more expensive than in IPv6.

Resolving the problem with NAT

Many of the same problems that exist today in the internet backbone are also being felt at the level of the enterprise and the individual business user. When an enterprise cannot summarize its routes effectively, it puts a heavy load on the global route tables. If enterprises do not receive sufficient globally unique address space, they are forced to deploy private, isolated address space that is not visible to the internet.

Since this private address space contains globally non-unique addresses, the users typically require gateways, and Network Address Translators (NATs) to manage their connectivity to the outside world. In such situations, some services are simply not available or only in a restricted way. A NAT allows an enterprise to have whatever internal address structure it desires, without concern for integrating internal addresses with the global internet. This is seen as particularly convenient in the existing IPv4 world, with its address restrictions. The NAT device sits on the border between the internet and the enterprise network, converting private internal addresses to a smaller pool of globally unique addresses that are passed to the internet backbone and vice versa.

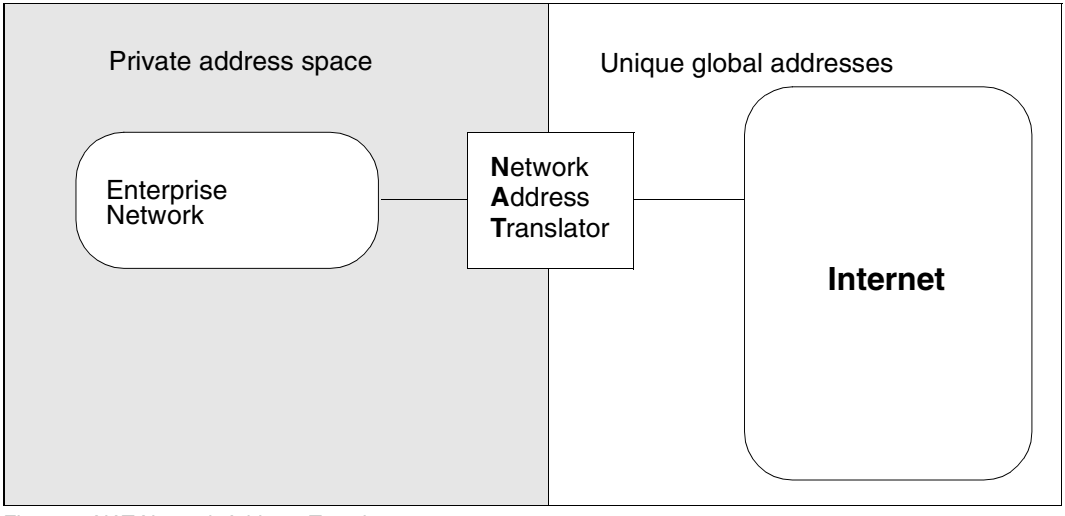


Figure 1: NAT Network Address Translator

However, NAT should only be used in organizations that do not require full connectivity with the outside world as NAT often stands more in the way of robust and efficient interaction with the internet. The NAT technique of substituting address fields in each and every packet that leaves and enters the enterprise is very demanding. As an enterprise's internet access increases, the NAT's performance must increase in parallel. The bottleneck effect is exacerbated by the difficulty of synchronizing access by various NAT devices to the enterprise network. Enterprises with NAT are less likely to achieve the reliable high-performance internet connectivity that is common today with multiple routers attached to the internet backbone. Furthermore, use of NAT devices detracts from the possibility for symmetric routing, since NAT devices require control of traffic in both directions between the enterprise network and the internet.

NAT translators also run into trouble when applications embed IP addresses in the packet payload, above the network layer. This is the case for a number of applications, including certain File Transfer Protocol (FTP) programs and Mobile IP. Although a NAT parses every packet all the way to the application level, it is likely to fail to translate some embedded addresses, which will lead to application failures.

NATs also prevent the use of IP-level security mechanisms. Today, NAT devices are helpful in certain limited scenarios for smaller networks, but are considered by many to be generally disadvantageous for the long-term development of the internet.

3.2.2 Minimizing administrative workload

The most time and cost-intensive part of today's network administration involves the assignment of networking parameters to computers and other network nodes that are needed before they can begin any sort of network operation. Information such as an IP address, DNS server, default router and other configuration details have to be installed in every active network component. In many cases, this is still done by manual configuration, either by the network administration, or worse still by the users themselves. Various efforts to shift this administrative load onto central servers has led to the development of the Dynamic Host Configuration Protocol (DHCP), but this brings its own administrative difficulties.

IPv4's limitations also aggravate the need in many organizations to assign new IP addresses to network devices. When an enterprise changes ISP, for example, it may have to either renumber all addresses to match the new ISP-assigned prefix or implement Network Address Translation devices (NATs). Renumbering may likewise be necessary when two companies merge or parts of an enterprise are spun off.

Routing prefixes are assigned to reflect the routing topology of the enterprise network and the number of connected nodes. There are two ways that the choice of routing prefixes can become inconvenient or incorrect:

1. The routing prefix can become too long which means that network administrators cannot connect any further network devices to special subnets and
2. The ways that the individual networks are connected together and to the outside world can change.

Either of these occurrence would indicate the need to renumber part or all of the enterprise network. However, it would be good to be able to renumber the enterprise network without requiring expensive downtime for the entire network or individual network components.

Address shortages and routing hierarchy problems threaten the network operations of larger enterprises, but they also affect small sites - even the home worker who dials in to the office via the internet may be affected.

Small networks can be completely dropped from internet backbone route tables if they are not adapted in accordance with their address structure. Because renumbering is no longer possible with larger networks, this causes serious problems for ISPs in ensuring access to the internet. With today's IPv4 address assignment, ISPs with many individual dial-in clients cannot allocate IP addresses as freely as they wish. Consequently, many dial-in users must use an IP address allocated from a limited pool on a temporary basis.

However, a unique IP address sets the stage for users to gain full connectivity to other users on the internet. It also simplifies a wide range of interactive production applications, of which remote diagnostic applications and telephoning via the internet are just two

examples. Today's hierarchy of limited and poorly allocated IPv4 addresses has already caused problems, and will continue to do so as more and more devices of varying capabilities are added to the internet.

3.2.3 Security

Encryption, authentication and data integrity safeguards are absolutely essential for commercial use of the internet. For these purposes, IPv6 offers security header extensions.

Authentication

The IPv6 authentication header allows a receiver to determine with a high degree of certainty that an IPv6 packet actually originated from the specified source address and that it was not changed on its route between sender and recipient. This prevents malicious users from changing your computer's IP address in order to disguise their true identity. Such source-address masquerading (spoofing) is among the techniques that could be misused in order to obtain confidential corporate information or to gain control of the corporate server. Spoofing might fool a server into granting access to valuable data, passwords, or network control utilities. IP spoofing is the best known and most widely used form of denial-of-service attack.

With IPv4 it is normally impossible for a server to determine whether packets are being received from the legitimate sender. Some enterprises have responded by installing firewalls, but these devices introduce a number of new problems, including performance bottlenecks, restrictive network policies and limited connectivity to the internet or even limited communication between divisions or sites of the same company.

IPv6 uses a standard method to determine the authenticity of packets received at the network layer, ensuring that network products from different vendors can use interoperable authentication services. IPv6 implementations are required to support the MD5 and SHA-1 algorithms for authentication and integrity checking to ensure that any two IPv6 nodes can interoperate securely. Since the header definition is independent of the authentication algorithm used, other authentication algorithms may be used as well.

Encryption

Along with packet spoofing, another major shortcoming in internet security is the misuse of network "sniffers" and other typically helpful diagnostic devices.

In IPv6, privacy (data confidentiality) is provided by an additional extension header for end-to-end encryption at the network layer. The IPv6 encryption header simply contains indicators as to the keys used. This means that the key information cannot be accessed illegally by analyzing the encrypted packets.

IPv4 network-layer extensions for this have been defined and are compatible with those for IPv6, but are not yet in wide use or are supported in the IPv4 stacks.

Both IPv6 security headers can be used directly for communication between two nodes as well as between special security gateways, which add an additional level of security to the IPv6 packets.

3.2.4 Mobility

IPv4 has difficulties managing mobile computers, such as notebooks, for several reasons:

– A mobile computer needs a forwarding address at each new point of attachment to the internet, and it's not always easy to get such an address with IPv4.

– Informing any agent in the routing infrastructure about the mobile node's new location requires good authentication facilities, which are not commonly deployed in IPv4 nodes.

– In IPv4, it may be difficult for mobile nodes to determine whether or not they are attached to the same network as before.

– It is most unlikely in IPv4 that mobile nodes would be able to inform their communication partners about any change in location.

The benefits for mobile computing are apparent in quite a number of aspects of the IPv6 protocol design, and go far beyond merely providing improvements in dialling in to the internet. The improvements in processing destination options, autoconfiguration, routing headers, encapsulation, security and anycast addresses all contribute to improved integration of mobile computing in an IPv6-based internet. The IPv6 mobility advantage may be further emphasized by combining flow label management to provide better Quality of Service to mobile nodes.

3.3 The IPv6 solution

Compared with IPv4, IPv6 offers a variety of key advantages:

– Multi-level, global and hierarchical routing architecture
– Address autoconfiguration
– Simplified IPv6 header format
– Multicast
– Anycast

3.3.1 Multi-level global and hierarchical routing architecture

IPv6, with its immensely larger address space, defines a multi-level hierarchical global routing architecture. Using CIDR-style address prefixes, IPv6 addresses can be allocated in a way that facilitates route summarization. The growth of route tables in the backbone routers can be restricted and better controlled as a result. The much improved availability of IPv6 address space eliminates the need for private addresses. ISPs will have enough addresses to also allocate a globally unique IPv6 address to smaller businesses and individual users who dial in to the internet.

3.3.2 Address autoconfiguration

Each IPv6 node initially creates a local IPv6 address for itself using "stateless" address autoconfiguration, not requiring any manually configured local data or external servers.

Stateless autoconfiguration

Stateless autoconfiguration makes it possible for the IPv6 node to configure its own globally unique IP6v addresses in cooperation with a local IPv6 router. Typically, the node combines its 48-bit or 64-bit MAC (i.e. layer-2) address, assigned by the equipment manufacturer, with a network prefix passed on by a local router. This keeps end user costs down as network administrators are not required to properly configure each workstation before it can be deployed. These costs are currently a fixed part of the installation costs in IPv4 for each device being connected. With the much reduced administration costs and the use of extremely low cost network components, new market possibilities can be created for use of embedded systems. This feature will also help when residential networks emerge as an important market segment.

DHCPv6

IPv4 networks often use the Dynamic Host Configuration Protocol (DHCP) to reduce the effort associated with manually assigning IP addresses and other network parameters. DHCP is termed a "stateful" address configuration tool because it maintains static tables that determine which IP addresses are assigned to newly connected network components. A new version of DHCP has been developed for IPv6 (DHCPv6) to provide similar stateful address assignment as may be desired by many network administrators.

DHCPv6 also assists with efficient reconfiguration in addition to initial address assignment, by addressing any defined group of clients using multicast addresses.

The autoconfiguration capabilities of IPv6 will benefit internetwork users at many levels. For example, when an enterprise is forced to renumber because of an ISP change, IPv6 autoconfiguration will allow new addresses to be assigned to nodes, without even requiring manual reconfiguration of workstations or DHCP clients.

This function also assists enterprises in keeping up with dynamic end-user populations. Autoconfiguration allows mobile computers to receive valid forwarding addresses automatically, no matter where they connect to the network.

3.3.3 IPv6 header format

IPv6 simplifies the basic header layout of an IP packet. Some of the IPv4 header options were dropped in IPv6 while others were moved to separate additional headers. The simplified header structure is expected to offset the bandwidth cost of the longer IPv6 address fields. The 16-byte IPv6 addresses are four times longer than the 4-byte IPv4 addresses, but as a result of redesigning the IP header, the total IPv6 header size is only twice as large and many processing aspects are substantially more efficient than in IPv4.

In initial considerations of IPv6, a number of other solutions offering variable address lengths were looked at; in the end, simplicity won out, partially because 128 bits offers a sufficiently large address space for the longer term. Additional work in IP header compression promises to reduce or perhaps even effectively eliminate any additional network load associated with the use of 128-bit addresses over low bandwidth links.

IPv6 encodes IP header options in a way that streamlines the forwarding process in comparison with IPv4. Optional IPv6 header information is conveyed in independent extension headers located after the IPv6 header and before the transport layer header in each IPv6 packet.

Most IPv6 extension headers are not examined or processed by routers (in contrast with IPv4). This enables a big improvement in the usability of optional IPv6 features, compared to IPv4, where IP options typically cause a major performance loss for the packet at every intermediate router.

IPv6 extension header are variable in length and can contain more information than the IPv4 options. New header options can also be introduced in a straightforward manner.

More details of the comparisons between the IPv4 and IPv6 headers are discussed in the chapter “The technical case for IPv6” (see page 31).

We will simply point out briefly here that extension headers have been specified for carrying explicit routing information, as well as for supporting mobile computing, authentication, encryption and fragmentation.

3.3.4 Multicast

Modern internetworks need to transmit streams of video, audio, animated graphics, news and other timely data to groups of functionally related but dispersed nodes. This is best achieved by network layer multicast. Typically, a server sends out a single stream of multimedia or time-sensitive data to be read by subscribers who have signed on for it. The sender and individual receiver thus form a multicast group in this context.

A multicast-capable network routes the server’s packets to each subscriber in the multicast group using an efficient path. Packets are only replicated if the recipient is in different subnetworks. As can be seen in figure 2 on the next page, a single packet from the source will be received by all the multicast group members. When there are multiple networks containing multicast group members, a packet distribution "tree" is created for the multicast group.

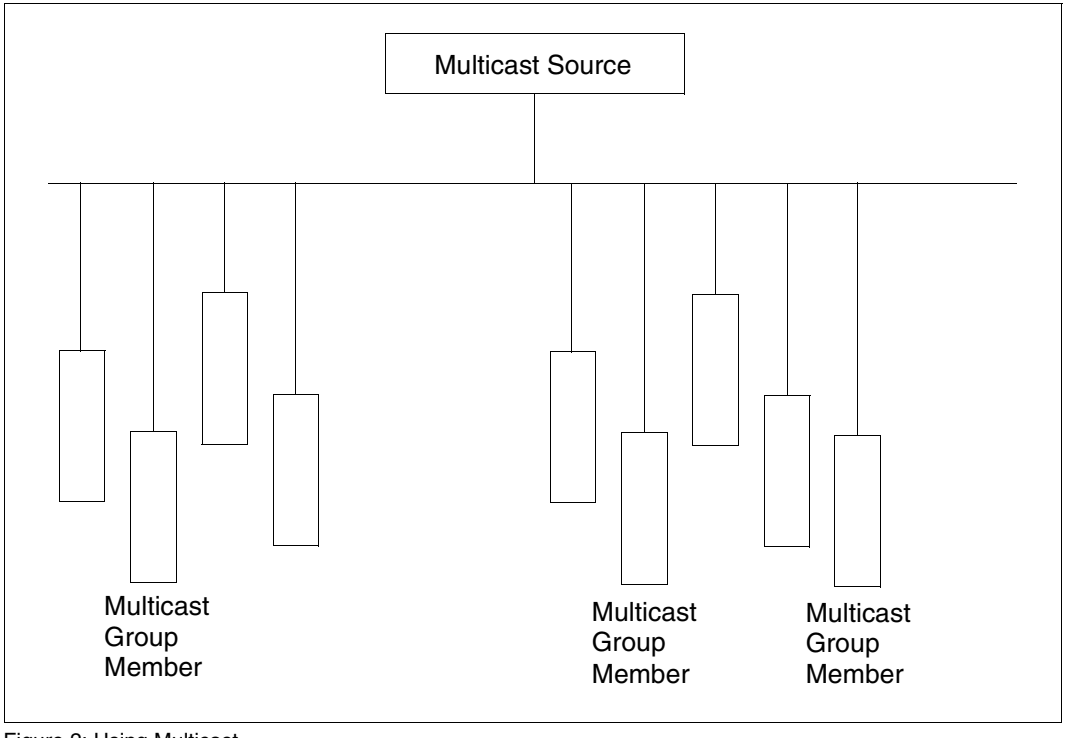


Figure 2: Using Multicast

Routers use multicast protocols such as Distance Vector Multicast Routing Protocol (DVMRP), Protocol Independent Multicast (PIM) or Multicast Open Shortest Path First (MOSPF), to dynamically construct the packet distribution tree that connects all members of a group with the multicast server. A new member becomes part of a multicast group by sending a "join" message to a nearby router. The distribution tree is then adjusted to include the new route.

Servers can then multicast a single packet, and it will be replicated as needed and forwarded through the internetwork to the multicast group. This conserves both server and network resources and, hence, is superior to unicast and broadcast solutions.

Multicast applications were developed initially for IPv4, but IPv6 extends IP multicasting capabilities by defining a much larger multicast address space. All IPv6 nodes and routers are required to support multicast. In fact, IPv6 offers the possibility of defining multicast addresses of different scope to replace the broadcasting capability no longer supported in IPv6.

3.3.5 Anycast

IPv6 anycast addressing is a new feature that does not exist in IPv4. Conceptually, anycast is a cross between unicast and multicast: an arbitrary collection of nodes may be designated as an anycast group. A packet addressed to the group's anycast address is delivered to only one of the nodes in the group. This is in contrast with multicast services, which deliver packets to all members of the multicast group. Nodes in an anycast group are specially configured to recognize anycast addresses, which are a subset of the unicast addresses.

Anycasting is a new service and its applications have not been fully developed. There are a wide variety of situations however, where anycasting makes sense.

Using anycast, an enterprise could forward packets to exactly one of the routers on its ISP's backbone. If all of a provider's routers have the same anycast address, traffic from the enterprise will have several redundant access points to the internet. And if one of the backbone routers goes down, another router in the anycast group automatically takes over its function.

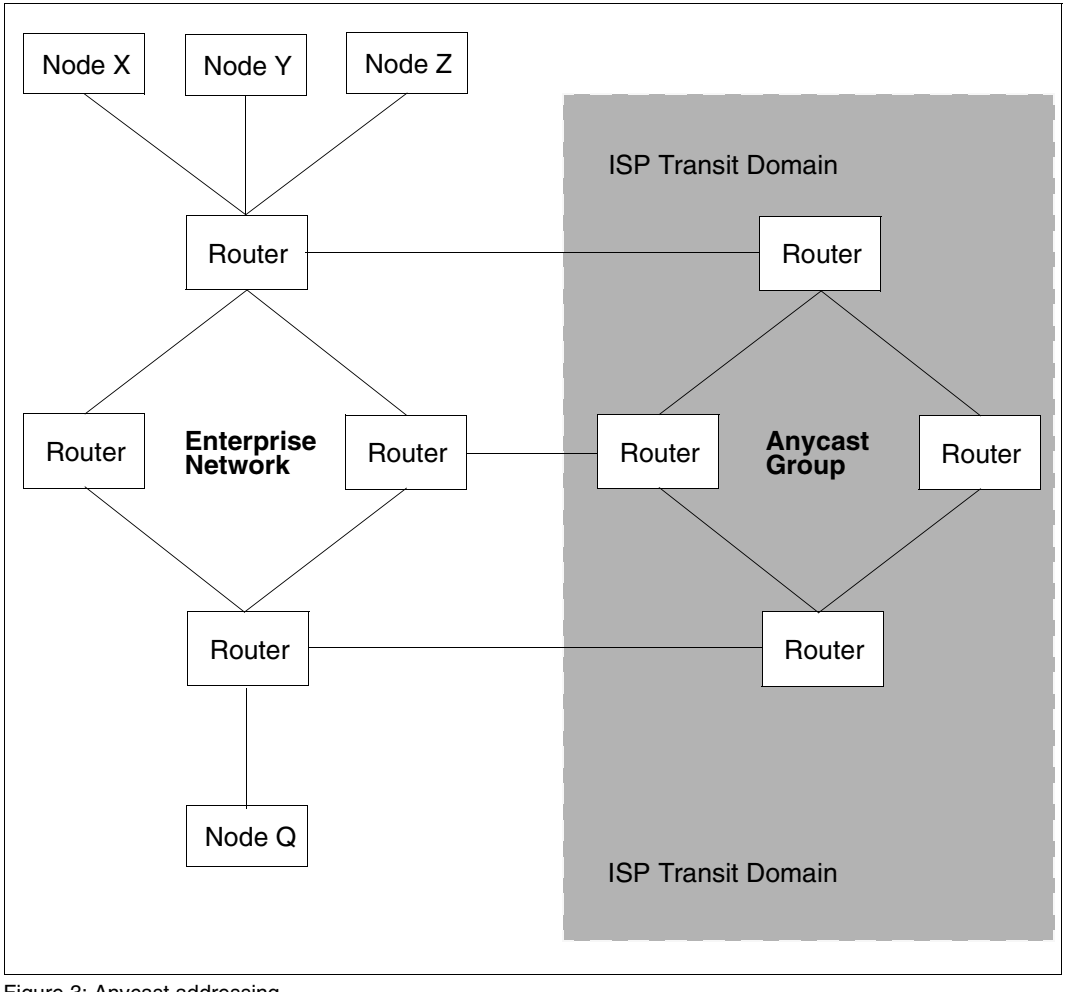


Figure 3: Anycast addressing

In figure 3, nodes Q, X, Y, and Z in an enterprise network send their packets to the anycast address served by the backbone routers in the ISP transit domain. The border routers in the enterprise network forward the packets just as they would for data sent to a unicast address. Then, any one of the backbone routers in the anycast group may receive the packets and forward them.

If the home agents for a mobile computer are likewise addressed via an anycast address, a redundancy concept can also be implemented here without additional configuration measures. In this case, each mobile computer works together with only one home agent even in the case when it doesn't know the address of any specific one.

Anycast was defined to allow nodes to efficiently and reliably access well-known services, mirrored databases, web servers etc. It can provide a cost-effective model for increasing the robustness of distributed applications and for performing load balancing. For instance, all the DNS servers in an enterprise can be assigned the same anycast address.

3.3.6 Quality of Service (QoS)

The IPv4 header contains a "differentiated services" byte, which corresponds to the "traffic class" byte in IPv6. Both are intended for support of simple differentiated services. Both IPv4 and IPv6 can support the RSVP protocol for more complex quality of service implementations. Additionally, the IPv6 header contains a 20-bit flow information field, which is available for use by additional quality-of-service functions.

Applications that use QoS functions are still in the planning stage, but IPv6 lays the foundation so that a wide range of QoS functions (including bandwidth reservation and delay bounds) may be made available in an open network environment.

An additional benefit for QoS functions in IPv6 is that the flow label has been allocated within the IPv6 base header and can therefore be used to distinguish traffic flows for optimized routing. Furthermore, the flow label can also be used even when encryption is used.

3.3.7 The transition to IPv6

The transition from IPv4 to IPv6 could take one of several paths. Some are lobbying for rapid adoption of IPv6 as soon as possible. Others prefer to defer IPv6 deployment until the IPv4 address space is exhausted, or until other technical issues leave no other choice. Either way, given the millions of existing IPv4 nodes and routers in the internet, IPv4 and IPv6 will have to coexist for an extended period of time

Therefore, designers have gone to great lengths to ensure that nodes and routers can be upgraded to IPv6 in a graceful, incremental manner. This smooth transition will prevent isolation of IPv4 nodes, and also prevent radical upgrades being needed from IPv4 to IPv6. Various transition mechanisms have been engineered to allow network administrators flexibility in how and when they upgrade nodes and routers. IPv6 can be deployed in nodes first, in routers first, or, alternatively, in an area with a limited number of nodes and routers. This area must neither comprise an adjacent subnetwork nor be restricted to one location.

Many upgraded nodes and routers will need to retain downward compatibility with IPv4 devices for an extended time period. For this reason, the upgraded devices will also retain their IPv4 addresses for some time to come.

To allow users upgrade smoothly from IPv4 to IPv6, suitable transition mechanisms were defined in a special IETF working group (NGTRANS). Examples here include dual-stack nodes and routers as well as tunneling of IPv6 packets over IPv4. A dual-stack node is a computer able to handle packets received and sent via both IPv4 and IPv6. Applications can then send and receive packets from both address families thus enabling easy transition from IPv4 to IPv6 at application level.

3.3.8 IPv6 DNS

Network administrators must consider conversions or extensions to their Domain Name Service (DNS) before deploying IPv6 nodes or dual-stack nodes. In response to this issue, a new DNS resource record type (AAAA) has been defined. However, during test operation of IPv6 in the 6Bone, weaknesses were discovered in the use of the AAAA DNS resource record and a new A6 DNS resource record type was defined for mapping DNS domain names to IPv6 addresses. The reverse mapping of IPv6 addresses to DNS names was also defined.

Once an IPv6-capable DNS server is in place, dual-stack nodes can interact interchangeably with IPv6 nodes. If a dual-stack node requests a partner IP address from the DNS server and receives back a 32-bit IPv4 address, it will communicate with the partner node on the basis of IPv4. If a 128-bit IPv6 address is received, then IPv6 is used. If there is no IPv6-capable DNS server, nodes can resolve name-to-IPv6-address mappings through the use of local name mapping tables.

IPv6 autoconfiguration and IPv6 DNS can be linked by using dynamic DNS updates. By these means, DNS servers can securely and automatically update their resource records whenever a new IPv6 node acquires a new IPv6 address. The automatic renumbering of IPv6 nodes is also supported in this way.

3.3.9 Modifying applications for IPv6

Applications that do not directly access TCP/IP network functions (i.e. do not call any SOCKETS functions) need no modifications to run in a dual-stack environment. In BS2000/OSD, in particular, this means that all DCAM-NEA, DCAM-ISO and CMX applications can run unchanged. This is also true if the IPv6 protocol is used at network level. SOCKETS applications that want to use IPv6 have to be modified to support IPv6. IPv6 is implemented at the SOCKETS interface by a new address family, which has to be used in this case. However, IPv6-enabled applications can also communicate with IPv4 partners so that two variants of the application, one for IPv4 and one for IPv6 do not have to run at the same time.

3.3.10 Routing in IPv6/IPv4 networks

Routers running both IPv6 and IPv4 can be administered in much the same fashion that IPv4-only routers are currently administered. Protocol extensions to BGP4 have been defined by the IETF to support IPv6. These extensions have been used widely in the 6Bone for IPv6 routing since early 1997. The BGP extensions have been implemented by all the major router vendors and are defined in an RFC. IPv6 versions of other popular routing protocols, such as Open Shortest Path First (OSPF) and Routing Information Protocol (RIP) have likewise been defined and implemented.

Network administrators may choose to keep the logical structure of the IPv6 network completely separate from the logical structure of their IPv4 network, even though both run on the same physical infrastructure, allowing the two to be administered separately. Alternatively, it may be advantageous to align the two logical structures by using the same domain boundaries and subnet organization. Both approaches have their advantages and disadvantages.

A separate IPv6 network structure can be used to replace the inefficient IPv4 network structures burdening many of today's enterprises and to introduce a new, structured, hierarchical network address plan that will provide optimum connection to one or more ISPs. This simplifies renumbering, route aggregation (summarization) and other goals of a hierarchical network topology.

Initially, many IPv6 nodes may have direct connectivity to each other only via IPv4 routers. Such nodes will exist in islands of IPv6 topology surrounded by an ocean of IPv4. Transition mechanisms were therefore developed that allow IPv6 nodes to communicate over intervening IPv4 networks.

The essential technique of these mechanisms is the tunneling of IPv6 packets via IPv4 networks. IPv6 packets are packed within IPv4 packets for this purpose. Tunneling allows the individual IPv6 nodes to take advantage of the existing IPv4 infrastructure without changing any IPv4 components in the network.

A dual-stack router on the "edge" of the IPv6 island simply inserts ("encapsulates") an IPv4 header in front of each IPv6 packet and forwards it as a native IPv4 packet. IPv4 routers forward this traffic without recognizing that an IPv6 packet is involved. On the other side of the tunnel, another dual-stack router "decapsulates" (removes the extra IP header) the IPv6 packet and routes it to the ultimate destination using standard IPv6. The encapsulation and decapsulation of IPv6 packets can likewise be handled by the nodes.

There are two types of tunneling:

– Automatic
– Configured

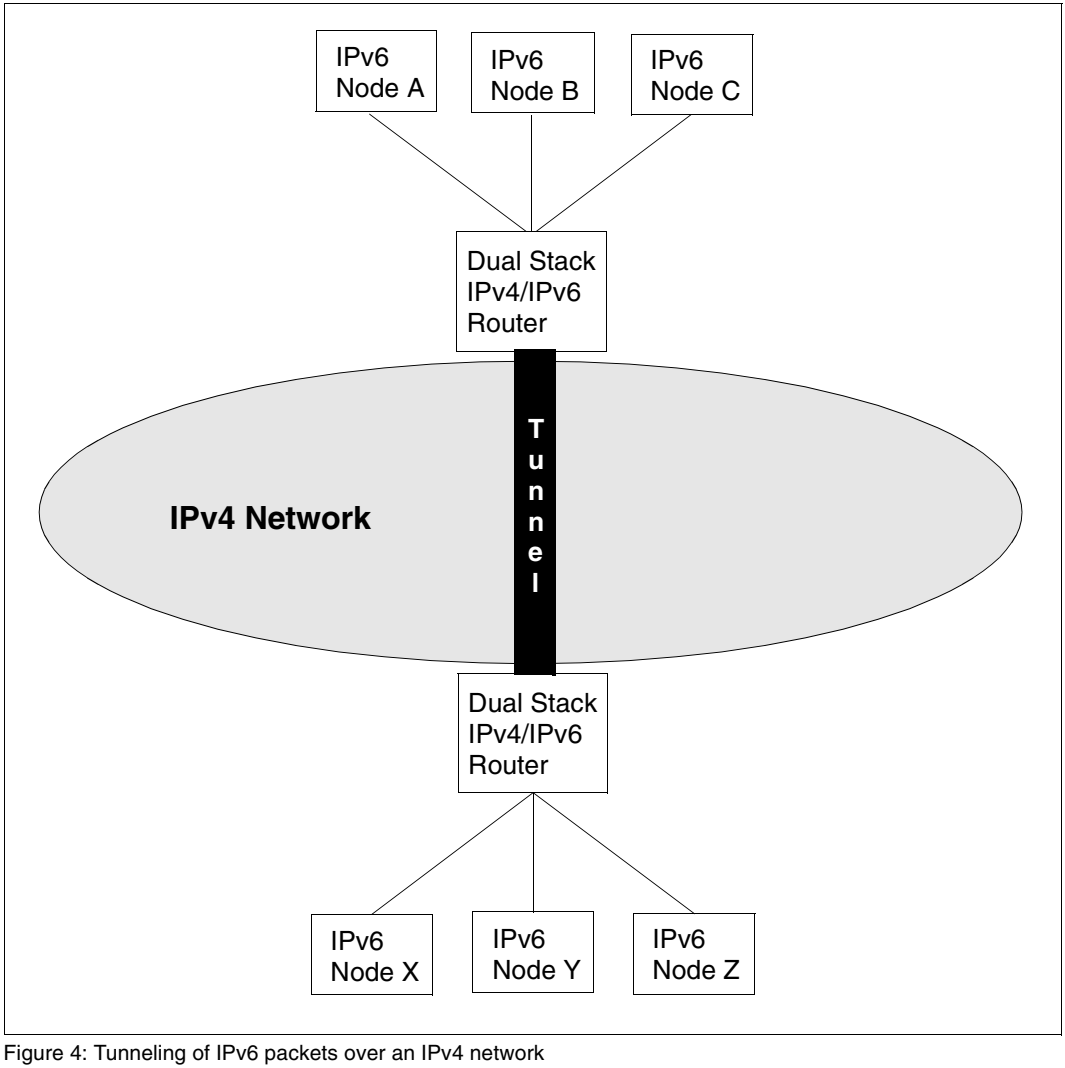


Figure 4: Tunneling of IPv6 packets over an IPv4 network

3.3.10.1 Automatic tunnels

Automatic tunnels use IPv4-compatible IPv6 addresses. An IPv4-compatible IPv6 address comprises an IPv4 address, which has been expanded by leading zeros to a 128-bit long IPv6 address.

When packets are forwarded with an IPv4-compatible IPv6 address, the device at the tunnel entry point can generate an IPv4 address from the IPv6 address and encapsulate the IPv6 packets in IPv4 packets. The IPv4 header is removed again at the other end of the tunnel.

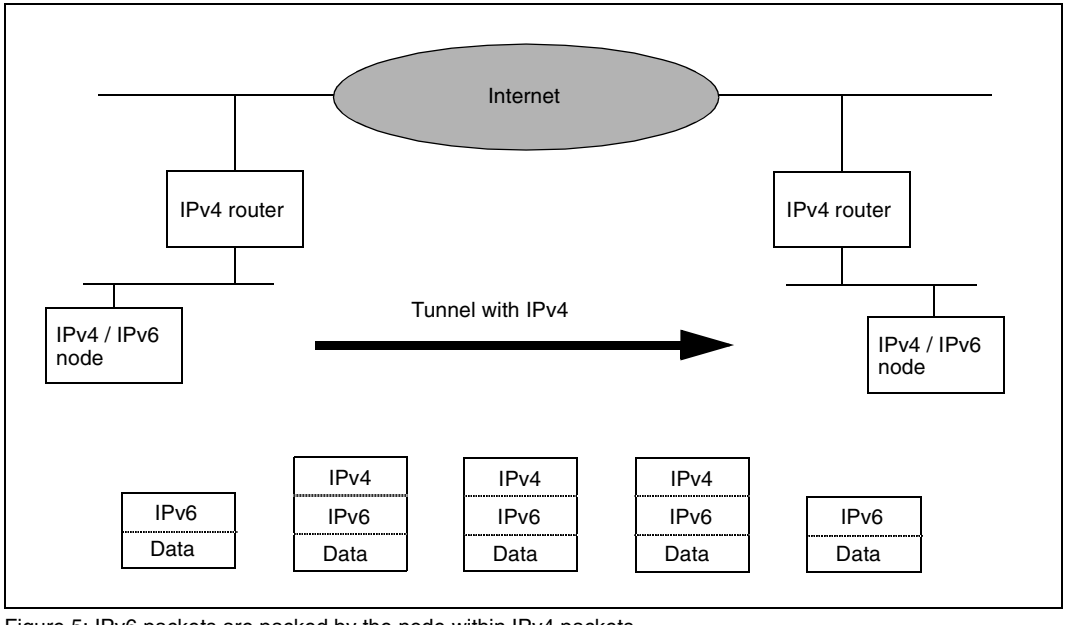


Figure 5: IPv6 packets are packed by the node within IPv4 packets

Automatic tunneling enables IPv6 nodes to use existing IPv4 networks in a simple and convenient way. However, it does require use of IPv4-compatible IPv6 addresses, which do not bring the benefits of the extended IPv6 address space.

IPv6 nodes using IPv4-compatible IPv6 addresses cannot take advantage of the extended address space, but they can exploit the other IPv6 enhancements, including flow labels, authentication, encryption, multicast and anycast.

Once a node is migrated to IPv6 with IPv4 compatibility, the door is open for a fairly painless move to the full IPv6 address space. IPv4-compatible addressing means that administrators can add IPv6 nodes to their network while initially preserving their basic address and network structure. Automatic tunnels are available when needed, but they may not be necessary when major backbone routers are upgraded to include the IPv6 stack. The upgrading of IPv4-compatible IPv6 addresses to normal IPv6 addresses can be achieved quickly and efficiently when backbone routers support IPv6 fully.

3.3.10.2 Configured tunnels

To build configured tunnels, network administrators must define IPv6-to-IPv4 address mappings at tunnel endpoints. Outside of the configured tunnel, IPv6 packets are forwarded with IPv6 addresses. At the entry point to the configured tunnel, the IPv6 packets are encapsulated in IPv4 packets and sent to the configured IPv4 destination address.

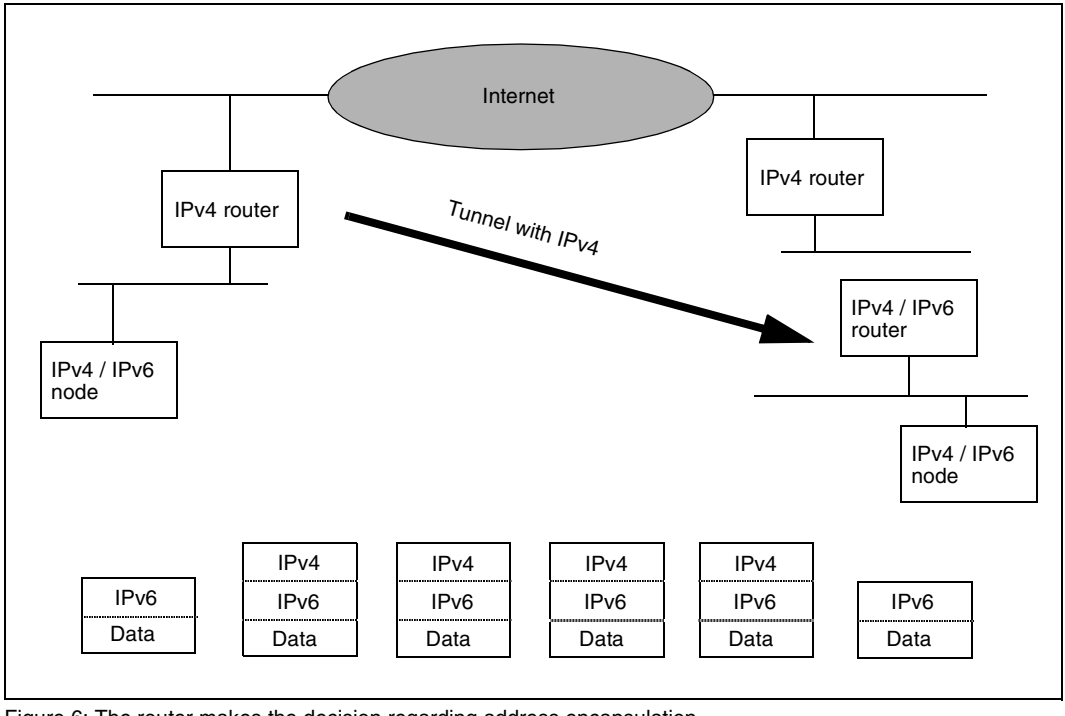


Figure 6: The router makes the decision regarding address encapsulation

The IPv6 packets are decapsulated again at the endpoint of the configured tunnel and forwarded normally using their IPv6 address. This requires a certain amount of manual administration at the endpoints of the configured tunnel, but the encapsulated IPv6 packets are routed through the IPv4 network dynamically, without the IPv4 routers noting the link with IPv6.

In contrast to automatic tunnels, there is no direct relationship between the IPv4 and IPv6 addresses in configured tunnels.

3.3.11 The dual stack transition method

Initial users of IPv6 nodes will require continued interaction with existing IPv4 nodes. This is accomplished with the dual-stack IPv4/IPv6 transition approach. Many nodes and routers in today's heterogeneous world of communication already support different network protocols. For instance, the majority of routers currently installed are multiprotocol routers and even many nodes support different network protocols. BS2000/OSD, for example, to date supported NEA, ISO and IPv4 protocols. The inclusion of one additional protocol on a node or router is therefore a well-understood problem. When running a dual IPv4/IPv6 stack on a node, this node has access to both IPv4 partners and IPv6 partners. An IPv4/IPv6 dual stack router can forward both IPv4 packets as well as IPv6 packets.

Dual stack machines can use both fully independent IPv4 and IPv6 addresses as well as IPv4-compatible IPv6 addresses.

3.4 Discussion on IPv6

Because of it potential for future dominance and the number of detailed technical choices that had to be made, IPv6 has attracted much controversy in the internet community. Such controversy has held back network owners who are in the process of crafting their forward-looking network strategy from implementing the measures necessary to upgrade to IPv6.

In the next sections, we try to counteract some of these myths and give the reader some arguments in favor of IPv6. Otherwise, there's a risk that the internet will not be able to progress beyond a patched-up version of IPv4.

The only driving force behind IPv6 is address space depletion

Many of the discussions about a new internet protocol focus on the fact that we will sooner or later run out of globally unique network layer addresses, due to IPv4's fixed 32-bit address space. The various address registries that assign blocks of IP addresses to large network service providers and network operators have become quite cautious about the way these addresses are handed out, because most predictions for IPv4 address exhaustion target a time frame within this decade.

With the long-haul in mind, IPv6 has been outfitted with a 128-bit address space that should guarantee globally unique addresses for every conceivable variety of network device for the foreseeable future (i.e. decades). IPv6 has 16-byte addresses, or 340.282.366.920.938.463.463.374.607.431.768.211.456 addresses (over a third of a duodecillion of them, in fact). The number of addresses gets a lot of attention but it is only one of many important issues that IPv6 designers have tackled. Other IPv6 capabilities have been developed in direct response to current business requirements for more scalable network architectures, mandatory security and data integrity, extended quality-of-service (QoS), autoconfiguration and efficient network route aggregation at the global backbone level. These features are all specified with IPv6 in a way that would be difficult to realize as effectively in IPv4.

Extensions to IPv4 can replicate IPv6 functionality

There have been multiple efforts to extend the life of IPv4 incrementally with evolutionary changes to the protocol standard and various proprietary techniques. One such example is the development of network address translators (NAT) that preserve IPv4 address space by intercepting data packets and converting private intra-enterprise addresses into one or a few globally unique internet addresses. Other examples include the various QoS and security enhancements to IPv4, which are in general scaled-back or identical to mechanisms specified in IPv6.

We do not know how long IPv4's life can be extended by these techniques. What is certain is that the widespread introduction of NAT devices negatively affects the end-to-end viability of familiar internet applications. In practice only a limited set of well-known applications can be correctly handled by NAT devices or by application level gateways associated with them. In particular, NAT devices restrict the deployment of end-to-end IPv4 security.

Furthermore, the development of new and innovative internet applications is burdened with the design constraints posed by NATs. Since NAT is strictly unnecessary for IPv6, standard end-to-end IPv6 security can be deployed. NAT translation is also known to create great difficulty in the construction of Virtual Private Networks (VPNs), since it makes address space administration difficult and interferes with standard security mechanisms.

NAT also only works in a "flat universe" for a site accessing the global internet. However, even moderately-sized enterprises are not flat internally, rather have nested multi-party relationships. Realistic NAT deployment solutions would have to include routing via multiple ingress/egress NATs for load balancing, multi-NAT-hop routes and so on. All this would create in miniature the IPv4 architecture, but piecewise and badly.

It is difficult to compare the costs of converting to IPv6 with those of remaining with IPv4 and its upgrades. Every network manager will have to make this comparison; but staying with IPv4 is not a satisfactory solution in the long term.

IPv6 support for a large diversity of network devices is not an end-user or business concern

Over the next few years, conventional computers on the internet will be joined by a myriad of new devices, including palmtop personal data assistants (PDA), hybrid mobile phone technology with data processing capabilities, smart set-top boxes with integrated web browsers and embedded network components in equipment ranging from office copy machines to kitchen appliances. Some of the new devices requiring IP addresses and connectivity will be consumer-oriented, but many will become integral to the information management functions of corporations and institutions of all sizes. These new devices require features not fully understood by most protocol designers during the initial growth of the IPv4 internet.

IPv6's 128-bit address space will allow businesses to deploy a huge array of new desktop, mobile and embedded network devices in a cost-effective, manageable way. Furthermore, IPv6's autoconfiguration features will make it feasible for large numbers of devices to attach dynamically to the network, without incurring unsupportable costs for the administration for an ever-increasing number of adds, moves, and changes.

The business requirement for IPv6 will be driven by end-user applications. Applications for mobile nodes, e-commerce etc. will be easier to design and implement using IPv6, especially as compared to IPv4 patched by NAT.

IPv6 is primarily relevant to backbone routers, not end-user applications

It is true that IPv6 address aggregation allows efficient multitiered routing hierarchies that prevent the uncontrolled growth of backbone router tables. But many of the advanced features of IPv6 also bring direct benefits to end-user applications at the workgroup and departmental levels. For instance, applications will have available the mandatory IPv6 encryption and authentication services as an integral part of the IP stack. For mobile business users and rapidly changing organizations, IPv6 autoconfiguration will allow the efficient assignment of IP addresses without the delays and cost associated with manual address administration or even traditional DHCP, which takes place in many current IP networks. IPv6 is very much both an end-user concern and a business concern. This concern will become increasingly important as QoS flows and QoS routing become important architectural components of the internet.

Asynchronous Transfer Mode (ATM) cell switching will negate the need for IPv6

ATM and other switching methods offer interesting technology for present and future internetworks, but ATM is, by itself, not a replacement for the packet routing internet architecture. ATM is better understood as a link-layer technology over an access medium. It gives some isolation properties and offers the promise of proven Quality of Service (QoS) for applications that need it. Even these hypothetical advantages are not yet fully developed for ATM, and it is possible that these advantages will be equally well available in future IPv6 networks not running over ATM.

Fortunately, network owners do not have to make a choice between ATM or IPv6 because the two protocols will continue to serve different and complementary roles in corporate networking. For many network designers, ATM is a useful transmission medium for highspeed IPv6 backbone networks. Standards and development work is being devoted to integrating ATM and IPv6 environments. IPv6, like its predecessor IPv4, provides network layer services over all major link types, including ATM, Ethernet, Token Ring, ISDN, Frame Relay, and T1.

IPv6 is something that only large telephone companies or the government should worry about

Some internet pundits have characterized IPv6 as a concern that's outside the enterprise network and outside the current time frame. In reality, IPv6 is a standards track and mainstream solution for the operation and continued efficiency of day-to-day business activities. But the only way that IPv6 will take hold and succeed is if businesses and institutions of all types come to terms with the inadequacies of IPv4 and begin to lay plans for migration. In the past few years, internet protocols have enabled a whole new style of distributed commerce that brings people together inside enterprises and gives enterprises access to the entire world. In fact, the sustained and impressive growth of the internet, which has inspired the current engineering efforts for IPv6, is in large measure due to the

penetration of the World Wide Web to business and end users. Offering services to such end users is of interest to many more institutions than merely governments and telephone companies.

IPv6 requires extensive modifications to existing operating systems, applications and programming techniques

IPv6 obviously requires certain modifications to the network protocol handling modules installed on the relevant computers. However, this typically requires little or no change to the base operating system. Support was provided for IPv6 in BS2000/OSD without changes outside of the BCAM transport system. Simple and natural modifications, typically confined to fewer than a dozen lines of the programs, can be made to enable applications to use IPv6 addresses directly. Since IPv6 reserves a part of its address space for compatibility with IPv4 addresses, applications modified to handle IPv6 addresses can still communicate with existing IPv4 clients and servers. Moreover, the transition strategies defined for IPv6 deployment within the IPv4 internet should make the gradual adoption of IPv6 a smooth process that allows existing applications to be converted for native IPv6 operation in a gradual, controlled manner.

IPv6 - Too Little, Too Soon

IPv6 appears as an incremental enhancement to IPv4, and some people say that if we are going to go to all the trouble to switch network-layer protocols, we really ought to go all out for some really futuristic feature-full new protocol. This argument ignores the following simple facts:

– The purpose of a network-layer protocol is to hook together networks, and

– IPv6 builds on the amazing success of IPv4, by not forgetting the successful parts and by repairing the known faults. This is far different than starting over again with something unknown and untested.

Those who claim that it is too early for IPv6 ignore the facts that existing solutions extending the life of IPv4 are clearly stopgap measures, and that one can put IPv6 into service now.

Renumbering is fixed in IPv6

Although IPv6 has gone a long way to enable more convenient renumbering operations, it is a mistake to say that renumbering is a completely solved problem. IPv6 engineers are still considering designs for renumbering routers. Furthermore, applications that have been ported from IPv4 to IPv6 do not automatically become more able to support renumbering. Some applications will require small design improvements in order to support renumbering. Lastly, the biggest impediment to renumbering seems typically to be the institution of administrative practices that apply key information directly on IP addresses instead of using some

more appropriate indexing method. These administrative practices require attention and adherence to more modern guidelines for internet administration before the problem of renumbering can be considered to be solved.

Routing is fixed in IPv6

IPv6 offers improvements for routing in a number of ways. It allows for allocation of IPv6 addresses in a way that is more favorable for aggregation than existing IPv4 allocations. It allows for more streamlined packet forwarding than IPv4 routers can do, especially when IP options are used. IPv6's larger address space offers opportunities for more optimal network planning, since the constraints for planning out network connectivity have been relaxed to such a great extent. Furthermore, since every IPv6 router can be presumed to have security processing enabled, it is much easier to institute the appropriate security measures for authentication and keeping private data private.

However, there are still many operational issues that need attention. IPv6 routing protocols are largely adapted from almost identical IPv4 routing protocols, and thus inherit some of the same problems. Improvements continue to be made to routing protocols to improve their stability, convergence time and configurability.

One of the hardest problems is to make routing protocols more human-friendly, so that it does not take a genius to make the routing fabric work reliably. There are remaining issues surrounding multi-homing that have not been solved. All of these issues will continue to receive the attention of engineers involved with the ongoing development of IPv6.

4 The technical case for IPv6

This chapter discusses the technical aspects of IPv6. In many cases, the technical details illustrate the concepts of the previous chapter. However, other more detailed features of the protocol are introduced as needed to help provide a fuller understanding of the technical side of IPv6.

4.1 IPv6 headers vs. IPv4 headers

To start the technical look at IPv6, we compare the IPv6 header with the IPv4 header. Both headers carry version numbers and source/destination addresses, but as figure 8 shows, the IPv6 header is much simpler than the IPv4 header, which makes for more efficient processing by routers and nodes. Whereas IPv4 headers are variable in length, IPv6 headers have a fixed length of 40 bytes. Additional processing efficiencies have been achieved by reducing the number of required header fields in IPv6. An IPv4 header, illustrated in figure 7 contains at least 12 fields and can also contain additional option fields, not illustrated in the figure. IPv6, on the other hand, only uses 8 fields (see figure 8 on the next page).

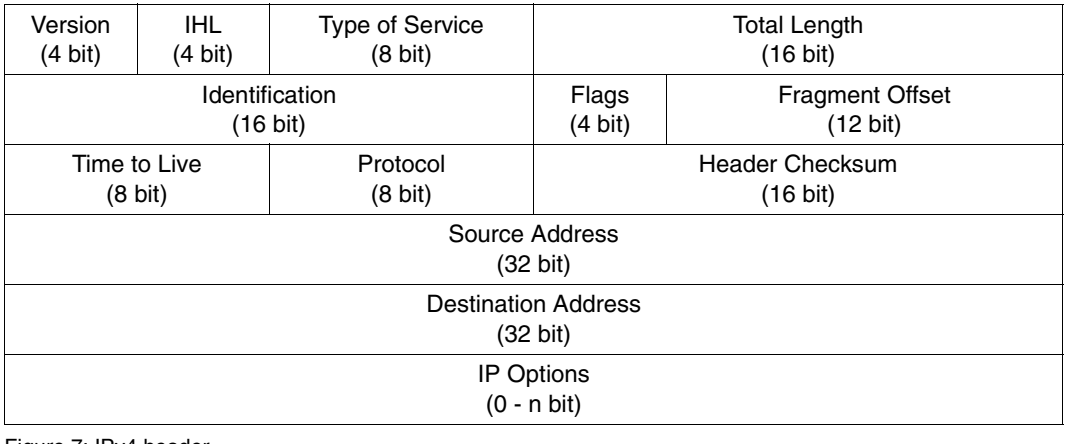


Figure 7: IPv4 header

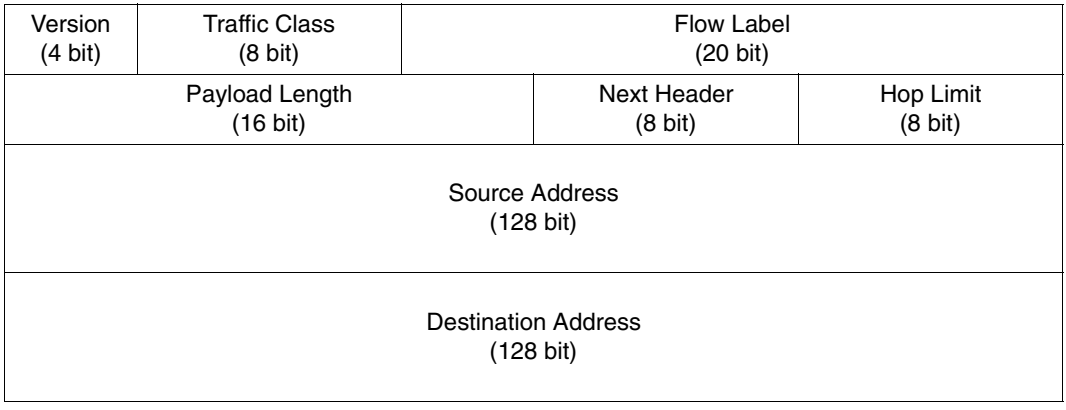


Figure 8: IPv6 header

The first field to be discarded in the IPv6 header is the header length field, which is clearly no longer required due to the fixed header length of all IPv6 packets. The total length field of the IPv4 header has been replaced by the IPv6 payload length field. But this field does not include the length of the IPv6 header, which is always assumed to be 40 bytes. The new payload length field can accommodate IPv6 packets up to 64 KB in length. Even larger IPv6 packets, called "jumbograms", can be passed between IPv6 nodes if the payload length field is set to zero and the length of the jumbogram is specified in a special extension header, as discussed below.

The time-to-live (TTL) field of IPv4 has been renamed the "hop limit" field in IPv6, to describe more accurately its actual function. The field is used to break routing loops. The hop limit is set by the sender of the IPv6 packet and decremented by one by each router that forwards the packet. When the value in the hop limit field is decremented to zero, the IPv6 packet is discarded. The IPv6 hop-count field allows up to 255 hops, which exceeds the needs for even the largest of networks, as calculated by IPv6 protocol designers.

In addition to the header length field, a number of basic IPv4 fields were eliminated from the IPv6 header. The fields "fragment offset", "identification", "flags" have been discarded as they were moved out to optional extension headers (see the section "Fragmentation headers" on page 35).

The IPv4 header checksum field has been abandoned in IPv6, since error checking typically is duplicated at other layers of the protocol stack. Bad packets will either be detected at the link layer or at the transport layer. It has been found in IPv4 that the calculation or checking of the header checksum reduces performance of nodes and routers considerably.

Finally, the IPv4 type-of-service field is replaced in IPv6 by the traffic class and flow label fields.

4.2 Extension headers

IPv4 headers include an options field, which conveys information about security, source routing and other optional parameters. Unfortunately, options are poorly utilized because routers typically offer degraded performance to packets that contain such options.

The IPv4 options field has been replaced in IPv6 by extension headers that are located after the primary IPv6 header and before the transport layer headers. IPv6 extension headers allow the use of security functions, fragmentation, source routing and other IPv6 functions. There is no protocol-dependent upper limit on the number of extension headers.Since IPv6 separates options into modular headers, they can now be processed more efficiently. Figure 9 shows encryption and fragmentation headers occurring after the primary IPv6 header and before the transport header.

IPv6 Hdr	Fragmentation Hdr	Encryption Hdr	Transport Hdr, etc.

Figure 9: IPv6 extension headers

The protocol field of the IPv4 header - normally Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) - has been replaced by the "next header" field. This header field indicates the type of the next header, which can be a TCP or UDP header or another IPv6 extension header.

IETF working groups have already defined a number of extension headers for IPv6 and have defined the ordering of headers, if they are present in a packet.

The suggested order for extension headers is as follows:

1. Primary IPv6 header
2. Hop-by-hop options header
3. Destination options header 1
4. Routing header
5. Fragmentation header
6. Authentication header
7. Encryption header
8. Destination options header 2

followed by the transport layer and payload headers.

Each extension header typically occurs only once within a given IPv6 packet, except for the destination options header.

4.2.1 Hop-by-hop options header

When present, this hop-by-hop options header carries information for all routers that forward the packet to the destination address. It must be the first extension header after the primary IPv6 header. Since this header is read by all routers along its path through the network, it is useful for transmitting debugging information to routers.

One currently defined application of the hop-by-hop options header is the router alert option, which informs routers that the IPv6 packet should be processed completely by a router before it is forwarded to the next hop. An example of such a packet is an RSVP resource reservation message for QoS services.

4.2.2 Destination options header

There are two variations of the destination options header, each with a different position in the IPv6 packet. A destination options header appearing before a routing header in the IPv6 packet will be processed by every router that forwards the packet. A destination options header appearing after a routing header in the IPv6 packet will only be processed by the destination system addressed. There are no actual applications implemented at present for use of the destination options header.

4.2.3 Routing header

IPv6 has to date defined a "Type 0" routing header, which gives the sending node a great deal of control over the IPv6 packet's route to the destination system. The IPv6 routing header replaces the loose source route (LSR) option in IPv4. This optional header allows a source node to specify a list of IPv6 addresses that the IPv6 packet has to pass on its way to the destination system.

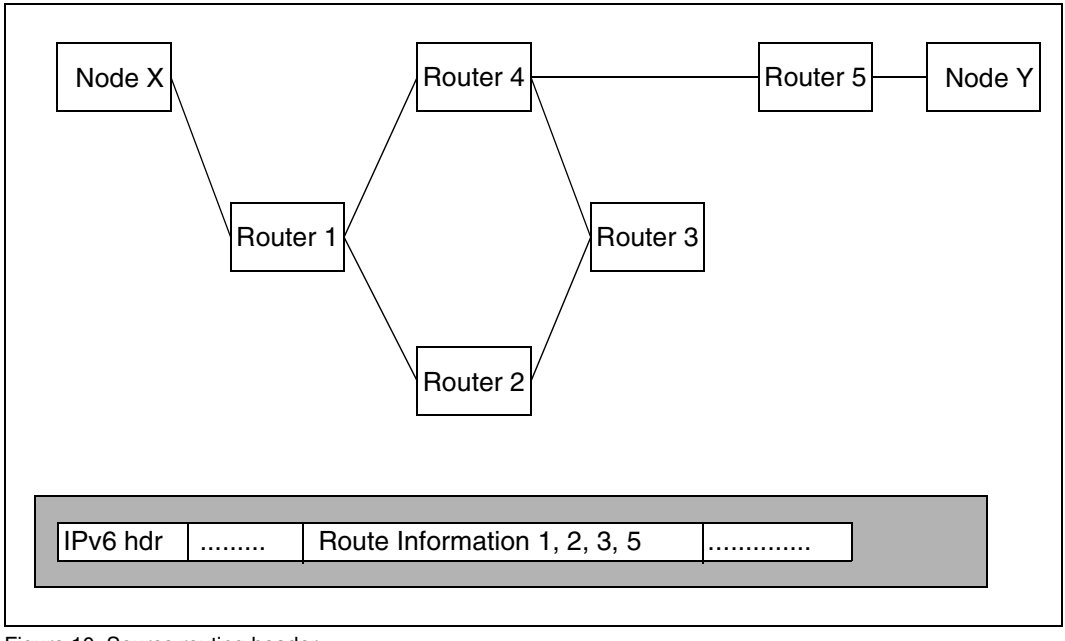


Figure 10: Source routing header

An example of IPv6's loose source routing (LSR) is illustrated in figure 10. In "loose" routing, routers not listed in the IPv6 routing header can also forward the IPv6 packet. So, for example in figure 10 the packet could be routed from router 3 through router 4 and then to router 5, even though router 4 was not specified in the routing information field of the IPv6 routing header.

The IPv6 routing header contains a counter for controlling which router is to be addressed next by the IPv6 packet. This counter shows the number of IPv6 routers not yet visited and each router specified in the list decrements the counter by 1.

4.2.4 Fragmentation header

The IPv4 network protocol has the ability to split or fragment IPv4 packets at any point in the path, depending on the transmission capabilities of the links involved. This feature has been dropped in IPv6 in favor of end-to-end fragmentation/reassembly between the source and destination nodes. Routers therefore no longer fragment large IPv6 packets.

The elimination of fragmentation by routers allows a simplified IPv6 packet header design and better router performance when forwarding IPv6 packets. Today's networks at link layer level generally support frame sizes that are large enough to carry typical IPv6 packets. In the event that fragmentation is required, IPv6 provides an optional extension header that is

used by source nodes to divide the original packet into smaller IPv6 packets. The IPv6 destination node will reassemble these fragments in a manner that is transparent to upper layer protocols and applications.

The IPv6 fragmentation header contains a field that identifies a group of IPv6 fragments as well as sequence numbers that define the sequence of the fragments.

The source node is responsible for sizing the individual IIPv6 packets correctly, so it has to determine the minimum size of the Maximum Transmission Unit (MTU) of the path to the destination system. This is the smallest MTU of a link on the route between the source node and the destination node. For instance (see figure 11) if two FDDI networks with a 4500-byte MTU are connected by an Ethernet with an MTU of 1500 bytes, then the source node must only send IPv6 packets that are no larger than 1500 bytes. Should a 2500 byte packet be sent, this will be split into two fragments, one 1500 bytes and one 1000 bytes in length.

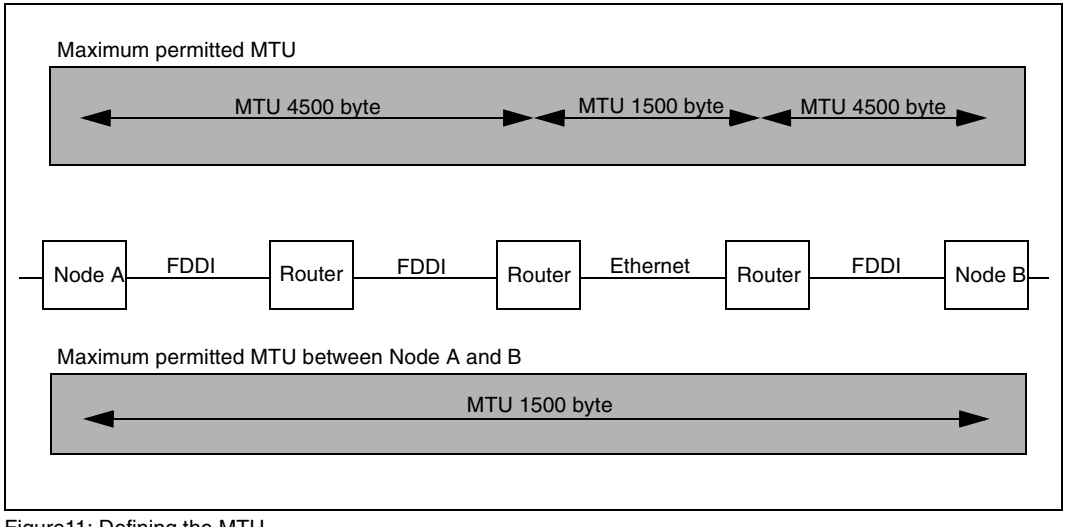


Figure11: Defining the MTU

End nodes can determine the MTU of a path with the MTU path discovery process. Typically, with this technique, the source node attempts to establish the MTU of the path to the destination system on a trial and error basis. First, a packet with the link-specific MTU of the source system is sent. If this MTU is too large for some link along the path, the router that detects this sends an ICMP "Datagram too big" message back to the source. This message also contains the MTU of the link for which the ICMP message was sent. The source can then adjust the packet size downward (fragment) and retransmit the smaller fragment. This process is repeated until an IPv6 packet gets all the way to the destination node. The discovered MTU is then used for fragmentation of IPv6 packets to the respective destination end node.

4.2.5 Extension header for secure data transfer

IPv6 offers two extension headers for ensuring secure data transfer:

– Extension header for authenticating incoming IPv6 packets
– Extension header for encrypting IPv6 packets

4.2.5.1 Authentication header

Using the IPv6 authentication header, nodes can verify the authenticity and integrity of the IPv6 packets received. The IPv6 authentication header makes use of an established security association, which may, for instance, be based on the exchange of secret keys. In a client/server application, for instance, both the client and the server need to have knowledge of the secret key. Before an IPv6 packet is sent, a Message Integrity Code (MIC) is created for the entire packet based on the key. The data from the IPv6 authentication header is added to the signature to eliminate replay attacks. The MIC is then recomputed on the receiving side and compared with the transferred MIC. If they do not match, the packet is discarded. This approach provides authentication of the sender and guarantees that data within the packet has not been modified or replayed by an intervening party.

Authentication can take place between peer partners or between clients and servers.

4.2.5.2 Encryption header

Authentication headers eliminate a number of host spoofing and packet modification attacks, but they do not prevent passive reading of IPv6 packets traversing the internet and corporate backbone networks. This protection is offered by IPv6 with its encryption header (Encapsulating Security Payload (ESP)). Packets protected by the encryption header can have very high levels of privacy and integrity - something that is not widely available with the current internet, except with certain secure applications (e.g. private electronic mail and secure HTTP Web servers). The encryption header provides encryption at the network layer, making it available to all applications in a standardized way.

The IPv6 encryption header is used,

– to encrypt the transport layer header with the subsequent payload or
– to encrypt the appropriate IPv6 header fields along with the payload.

Both of these methods are accomplished with the encryption header, which ensures end-to-end encryption. When just the transport layer header and subsequent payload is to be encrypted, the encryption header is inserted in the packet directly before the transport layer headers (UDP, TCP). This is referred to as IPv6 encryption transport mode (see figure 12).

unencrypted			encrypted
IPv6 hdr	extension header	ESP hdr	transport hdr & payload

Figure 12: IPv6 encryption transport mode

If an even higher level of security demands encryption of the entire IPv6 packet, the packet is encapsulated as payload in a new IPv6 packet with an encryption header. This type of full IPv6 packet encryption including the original address information is referred to as IPv6 encryption tunnel mode (see figure 13).

unencrypted			encrypted			
IPv6 hdr	ext. hdr	ESP hdr	IPv6 hdr	ext. hdr	ESP hdr	transport & payload
additional encryption header			original packet			

Figure 13: IPv6 encryption tunnel mode

Fully encrypted IPv6 packets are somewhat more secure than IPv6 packets encrypted in transport mode because the original IPv6 header is no longer available for traffic analysis.

There is a considerable performance penalty for full encryption, due to the overhead and processing cost of creating an additional IPv6 header. In spite of its cost, tunnel mode encryption is particularly valuable to create a security tunnel between the firewalls of two remote sites (see figure 14). The fully encrypted IPv6 packet ensures that the original address information and any IPv6 extension header that may exist will not be visible. Within the tunnel, only the header of the additional IPv6 header required temporarily is visible. Once through the tunnel and safely within a firewall, the additional IPv6 header is stripped off and the packet is forwarded to its destination node using its own address information.

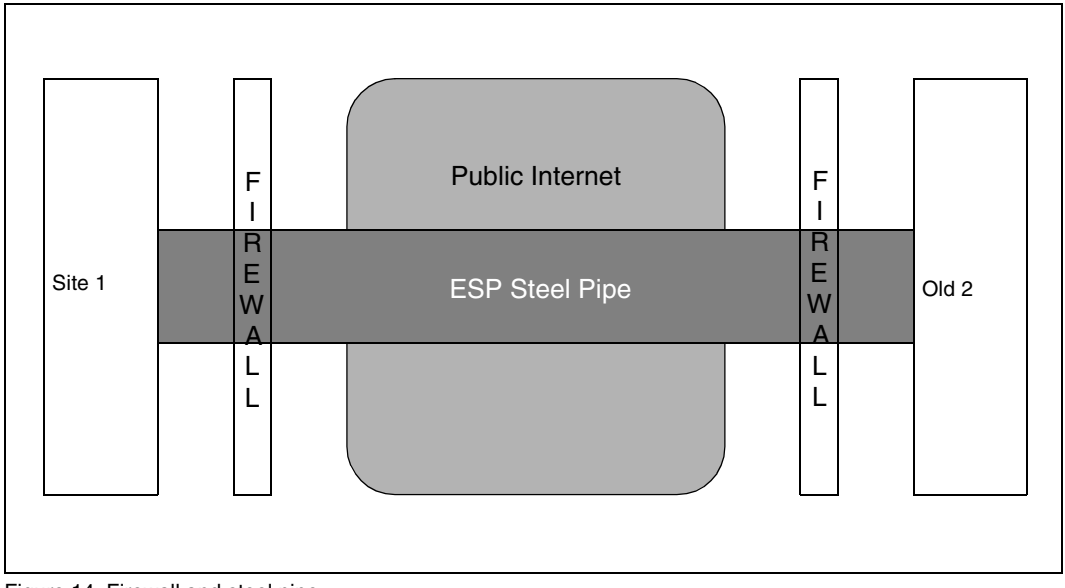


Figure 14: Firewall and steel pipe

4.2.5.3 Security solution

The encryption and authentication services of IPv6 together create the security solution that is being increasingly demanded in today's internet in the era of e-commerce. An authentication header is typically carried inside an encrypted datagram in tunnel mode, providing an additional layer of data integrity and verification of the sender's identification. In other cases, the authentication header may be placed in front of the encrypted transport mode portion of the packet. This approach is desirable if sender authentication takes place before decryption on the receiving end.

Taken together, the authentication and encryption services of IPv6 provide a robust, standards-based security mechanism that will play a decisive role in the continuing expansion of the internet.

4.3 IPv6 address architecture

Much of the discussion of IPv4 versus IPv6 focuses on the relative size of the address fields of the two protocols (128-bit in IPv6 versus 32-bit in IPv4). But an equally important difference is the relative abilities of IPv6 to provide a hierarchical address structure that facilitates efficient routing architectures.

4.3.1 IPv4 address hierarchy

IPv4 was originally defined with class A, class B and class C addresses.

Bit 1	**8**	**9**	**16**	**17**	**24**	**25**	**32**	**Address type**
0	Network ID		Node ID					**class A**
10	Network ID			Node ID				**class B**
110	Network ID					Node ID		**class C**

Figure 15: IPv4 address types

This divided the IPv4 address into a network and node part but did not create a hierarchy that would allow a single high-level address to represent many lower-level addresses. Hierarchical address systems work in much the same way as telephony country codes or area codes, which allow long-haul phone switches to route calls efficiently to the correct country or region using only a portion of the full phone number.

As the internet grew, the non-hierarchical nature of the original IPv4 address space proved inadequate. This problem has been improved by use of CIDR (see page 8), but legacy IPv4 address assignments still hamper routing within the internet. These legacy assignments limit both local and global levels of internetworking.

To combat deficiencies at the local area network level, the IPv4 subnetting technique was developed, which allowed a more manageable division of large networks. Using IPv4 subnets, a single network address or the network part of an IPv4 address can stand for a number of physical networks, a technique that conserves a considerable number of IPv4 addresses. For example, a single Class B address can be used to access hundreds of physical networks, each of which itself could have hundreds of individual nodes.

At the level of large internet backbones and global routing, IPv4 addresses can be more efficiently aggregated with supernetting, a form of hierarchical addressing. With supernetting, backbone routers store a single address that allows packets to be forwarded to a

number of lower level networks. This can considerably reduce the size of route tables in backbone routers, which increases routing performance and lowers the amount of memory required by the routers for managing the route tables.

Subnetting and supernetting have been particularly useful in extending the viability of the IPv4 Class C addresses. Both of these techniques allow routers to create an address hierarchy for forwarding IPv4 packets.

The process of creating an IPv4 routing hierarchy was formalized in CIDR. For instance, CIDR allows a number of (plentiful) Class C addresses to be summarized by a single prefix address, allowing Class C addresses to function in a similar way to hard-to-get Class A and Class B addresses. CIDR has extended the life of IPv4 and helped the internet scale to its current size, but it has not been implemented in a consistent way across the internet and enterprise networks. Consequently, the routing efficiencies and IPv6 address conservation advantages of CIDR are not fully realized today, nor will they ever be fully realized due to the legacy nature of IPv4 networks and the difficulty of restructuring the IPv4 addresses. IPv4 will continue to waste a larger proportion of its address space and to burden routers with inefficient and excessively large route tables.

At the enterprise level of internetworking, IPv4 engenders a high administrative workload associated with maintaining subnet bit masks and node addresses within the subnet structure, particularly where there are large, dynamic populations of end users. When an end user is moved in the subnetting environment of an enterprise, for example, the move must be planned and implemented very carefully to ensure that the user is not decoupled from the enterprise network for too long.

4.3.2 The IPv6 address hierarchy

Motivated by the experience gained from IPv4, IPv6 designers made sure from the very beginning to provide a scalable address space that can be partitioned into a efficient global routing hierarchy. At the top of this hierarchy, address areas are defined as Top Level Aggregators (TLA). Further address areas are defined as Next Level Aggregators (NLA) within the address areas defined by the Top Level Aggregators. These Top Level Aggregators and Next Level Aggregators are assigned to individual ISPs or to international companies and organizations. Individual addresses or also address areas are assigned to the end user or customer within the Next Level Aggregator. This structure allows efficient routing in the backbone area because only one Top Level Aggregator entry is required in the backbone routers for all Next Level Aggregators within a Top Level Aggregator (see figure 16).

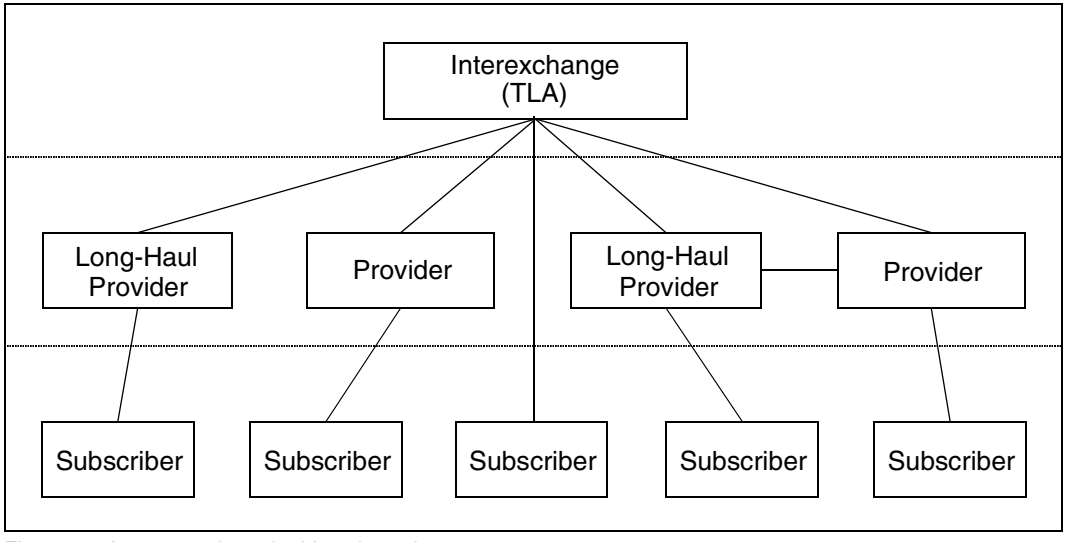


Figure 16: Aggregator based addressing schema

Although a number of allocation schemes are possible within IPv6's huge address space, an aggregation-based hierarchy was selected since it allows the greatest flexibility when assigning addresses. Provider allocation does not take account of the geographical circumstances of the network infrastructure, while allocation of addresses on a purely geographic basis (as with the telephone network), often conflicts with the requirements of large service providers.

An aggregation-based address structure considers the topological structure of the internet with a limited number of exchange points, where large long-haul service providers and telephone networks interconnect. The use of this topology by IPv6 addresses thus involves both a geographical component as well as a provider orientation, which takes account of the structure of the ISPs.

3 bit	13 bit	8 bit	24 bit	16 bit	64 bit
FP	TLA ID	reserved	NLA ID	SLA ID	Interface ID
Public Topology				Side Topology	Interface Identifier

Figure 17: Aggregator-based IPv6 addresses

Figure 17 shows the structure of an IPv6 address. The first 3 bits indicate what type of address follows (unicast, multicast, etc.). The next 13 bits make up the Top Level Aggregator ID (TLA ID). Eight bits are reserved for future use, the following 24 bits are allocated to the Next Level Aggregator Identifier (NLA ID) and finally 16 bits for the Site Level Aggregator Identifier (SLA ID).

The service providers can divide the NLA address field to create their own address hierarchy (see figure 18).

<table>
<tr><td colspan="4">24 bit</td><td>16 bit</td><td>64 bit</td></tr>
<tr><td>NLA 1</td><td colspan="3">Site</td><td>SLA</td><td>Interface ID</td></tr>
<tr><td></td><td>NLA 2</td><td colspan="2">Site</td><td>SLA</td><td>Interface ID</td></tr>
<tr><td colspan="2"></td><td>NLA 3</td><td>Site</td><td>SLA</td><td>Interface ID</td></tr>
</table>

Figure 18: Dividing the NLA address space

Typically, service providers supply their customers with blocks of contiguous IPv6 addresses, which are then used by individual customers to create their own local address hierarchy. The SLA field supports up to 65,535 individual subnets, each of which can contain 2^{64} different nodes. This 64-bit interface ID is used to identify the individual IPv6 interfaces. It is typically derived from the installed link layer address (MAC address) of the interface but can also be generated separately.

Internet backbone routers currently maintain up to 40,000 different routes. Because the internet is continuing to grow in size, IPv6's hierarchical address structure is the only viable method for keeping the size of backbone router tables under control. With an aggregator-based IPv6 address hierarchy, all internal networks at any hierarchical level can be reached by the relevant backbone routers using a single routing entry at a higher hierarchical level. This means that only the Top Level Aggregator part of the IPv6 address needs to be looked at to forward IPv6 packets at the highest hierarchical level of the internet backbone router.

IPv6's large hierarchical address space also allows a more decentralized approach to IP address allocation. Service providers can allocate addresses within address areas administered by them independently of central authorities, eliminating bureaucratic bottlenecks in the assignment of addresses as is currently the case due to the lack of IPv4 addresses.

Aggregation-based IPv6 addresses are just part of the total address space that has been defined for IPv6. Other address ranges have been assigned to multicasting addresses as well as for local site and link addresses.

Site-local and link-local addresses are available as private local addresses, which are only valid within a site or link and which do not have to be registered officially. IPv6 packets with site-local or link-local addresses are not forwarded outside the site or link. For example, this enable two separate companies to use the same site-local or link-local addresses without ever giving rise to conflict in the unique addressing of the nodes.

Site-local and link-local addresses have a major advantage when used for internal communication within a company. In this case, internal communication is not affected by switching service provider rather only cooperation with external partners.

Link local IPv6 addresses operate only over a single link, in other words they are not forwarded by routers. They can be used for temporary "bootstrapping" of network nodes before they receive a globally unique IPv6 address (see the next section). However, they can also continue to be used following bootstrapping.

4.4 Node address autoconfiguration

IPv6 has a large enough address architecture to accommodate internet expansion for many decades to come. Furthermore, IPv6 nodes can have their addresses automatically configured and reconfigured in a cost-effective and manageable way. Automatic address configuration is necessary in hierarchical routing because it supports scalable (and thus cost-effective) numbering and renumbering of large populations of IPv6 nodes. Even if the cost of assigning an address or renumbering a node is low, this still adds up to a major administrative overhead for service providers and large companies. However, if the overall costs for renumbering are low, a switch to another service provider may prove quite cost effective for a company even though this may not actually be feasible at present because of the high conversion overhead.

Autoconfiguration capabilities are important regardless of which style of address allocation is in effect. Occasionally, it may be necessary to renumber every node within an organization, as would be the case with a company that relocated its operations (with geographic addressing) or changed to another service provider (with provider-based addressing).

Configuration of IP addresses is a fact of life at the workgroup and department levels of large networked organizations. IP addresses need to be configured for new nodes, for nodes that change location or for nodes affected by the physical changes in the networks. In addition to these more traditional requirements for configuration by network administration, new requirements are emerging as large numbers of nodes become mobile.

These requirements for simple dynamic configuration of routers and nodes are basically not met with the IPv4. This is another reason why the concept of address autoconfiguration was developed in IPv6.

The process of address autoconfiguration under IPv6 starts with the Neighbor Discovery (ND) protocol.

The Neighbor Discovery Protocol combines and refines the services provided in the IPv4 environment by the Address Resolution Protocol (ARP), the Internet Control Message Protocol (ICMP) and Router Advertisement. The Neighbor Discovery Protocol is based on IPv6 extensions of the Control Message Protocol. The new ICMP messages allow routers and nodes to exchange both addresses and other network parameters at the same link.

In a typical scenario, a node starts the process of address autoconfiguration following connection to the network by creating a link-local address for the network interface.This address is typically derived from the node's IEEE interface address. Once this address is formed, the node sends out an ND message to ensure that the address is unique:

– If no ICMP Neighbor Solicitation message comes back, the address is presumed unique.

– If an ICMP Neighbor Solicitation message comes back, the address is not unique and the connected node must make another attempt with a different address.This will continue to happen until the node has a unique link-local IPv6 address.

Using the new link-local IPv6 address as a source address, the node then sends out an ICMP message "Neighbor Discovery Router Solicitation". The destination address used is an IPv6 multicast address, which can be used to address all routers on the link. The routers respond to the ICMP message "Neighbor Discovery Router Solicitation" with an ICMP message "Neighbor Discovery Router Advertisement", which apart from other data also contains the address prefix of the local network. The newly connected node can however also decline to send an ICMP message "Neighbor Discovery Router Solicitation" and wait for one of the ICMP messages "Neighbor Discovery Router Advertisement" sent periodically by the routers. Figure 19 shows ICMP messages being exchanged between the new node X and the router.

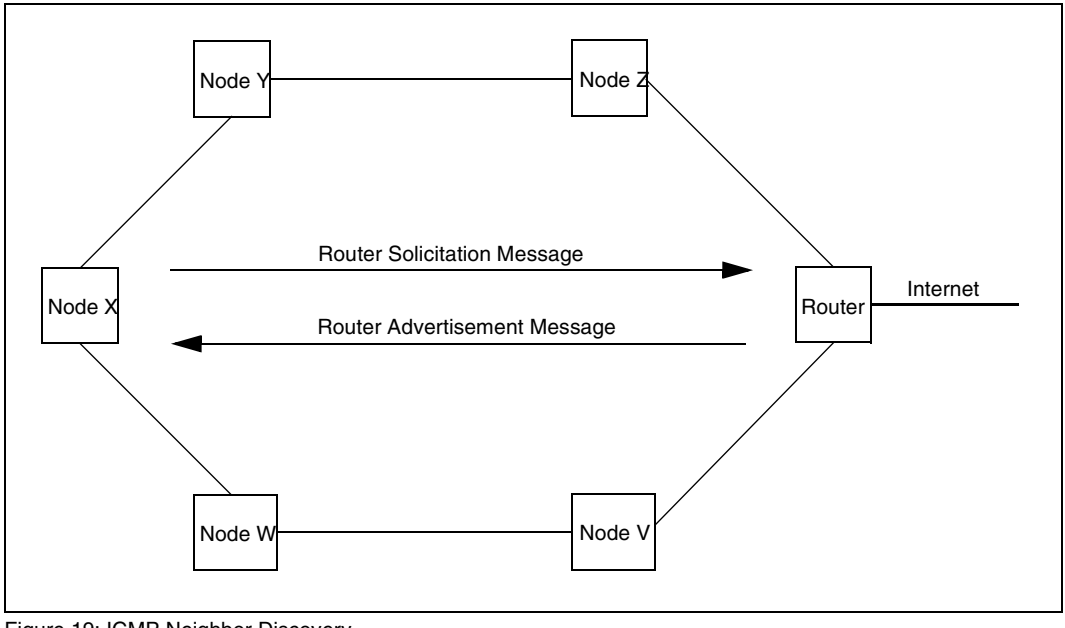


Figure 19: ICMP Neighbor Discovery

The ICMP message "Neighbor Discovery Router Advertisement" determines whether the newly connected node will perform a stateful or a stateless address autoconfiguration for the interface.

In the case of stateful address autoconfiguration, the node will contact an address server, which will assign it its global address. DHCP is the protocol of choice for autoconfiguration in IPv4 networks and has been reformulated for the IPv6 environment.

With the stateless approach, the new node can automatically configure its own global IPv6 address without the help of an address server or any human intervention. The node uses the globally valid address prefix information in the "Neighbor Discovery Router Advertisement" message to create its own IPv6 address. Together with the node part of the link-local address, this address prefix forms a universally unique global address.

If a node has several interfaces, this process of address autoconfiguration is performed for each interface.

The advantages of stateless address autoconfiguration are many. For instance, if an enterprise changes service providers, the prefix information from the new provider can be propagated by its routers to all nodes throughout the enterprise. Hypothetically, if all nodes in the enterprise use IPv6 stateless address autoconfiguration, the entire enterprise could be renumbered without any manual intervention on the nodes. At a more modest level,

workgroups with substantial move/change activity also benefit from stateless autoconfiguration because nodes can receive an automatically configured and valid IPv6 address each time they connect and reconnect to the network.

Address autoconfiguration plays an essential role in the support for mobile nodes. Each mobile node can be assigned a valid IPv6 address, no matter which network it is attached to. This address is used as a kind of forwarding address and is referred to as a "care-of address". The mobile node can receive all of its data from its home network by asking a router to forward packets to it at its care-of address. Better yet, the mobile node can also instruct any other node to forward data to its care-of address, so that the data never traverses the home network.

However, the mobile node is still identified by its home address, even though it is receiving packets sent to its care-of address. This is important to ensure that the node can be identified uniquely and also to enable a change of site during operation as is the case typically with a wireless connection.

To facilitate dynamic node renumbering, IPv6 has a built-in mechanism to create a graceful transition from old to new addresses. Fundamental to this mechanism is the ability of IPv6 nodes or routers to support multiple IPv6 addresses per physical interface. IPv6 addresses assigned to an interface can be identified as valid, deprecated or invalid. In the renumbering process, an interface's IPv6 address would become deprecated when a new address was automatically assigned. For a period of time after the new (valid) address is configured, the deprecated address continues to send and receive IPv6 traffic. This allows applications based on the older address to be finished gracefully. Eventually the deprecated IPv6 address becomes invalid and can no longer be used for communication. The new address is then used exclusively. Issuing multiple IPv6 addresses for an interface allows renumbering to occur dynamically and transparently to end users and applications. Besides simplifying renumbering of nodes, IPv6 has work underway to help with automatic renumbering of routers.

The stateless autoconfiguration process described above is particularly suited to conventional IP/LAN environments with 48-bit or 64-bit addressing at the link layer as well as hardware support for multicasting. Other network environments with different link layer characteristics may require modified or even alternative configuration techniques. For instance, current ATM networks do not inherently support multicast services. Research is underway to adapt the IPv6 Neighbor Discovery protocol to the special features of ATM networks as well as other network technologies.

4.5 Other protocols and services

The preceding discussion focuses on some of the more innovative and radical changes that IPv6 brings to internetworking. In many other areas, protocols and services will operate much the same as they did previously in the IPv4 world. As the internet moves to IPv6, PPP, DHCP and DNS servers are being modified to accommodate 128-bit IPv6 addresses, but in terms of basic functionality of these protocols, there will be little change. This is also generally true for the different routing protocols.

For example, a modified OSPF protocol is available with full support for IPv6, allowing routers to be addressed with 128-bit IPv6 addresses. The 32-bit link-state records of IPv4 OSFP will be replaced by 128-bit records for IPv6. In general, the OSPF data structures for IPv6 will run in parallel with the OSPF data structures for IPv4 topologies, so that the two data structures and the forwarding of IPv4 and IPv6 packets may coexist in the same router without major interaction. Due to the minor changes in the OSPF protocol for IPv6, network administrators with experience of OSFP for IPv4, should also have no problem when using the OSPF version for IPv6. A suitable update from IPv4 to IPv6 is also defined for the RIP routing protocol.

As with the interior gateway protocols, RFC 2545 is defining an IPv6-compatible version of an exterior gateway protocol, which will be used by the routers to guarantee accessibility of ISPs, large enterprises and other organizations via the internet backbone. The IPv4-based backbone routes use the Border Gateway Protocol (BGP), to disseminate CIDR-based routing information via the internet. Enhancements for an IPv6-based BGP are currently being defined.

5 Converting from IPv4 to IPv6

The implementation of an IPv6 network is comparable to the implementation of an IPv4 network. In both cases address space needs to be made available, which is used by the respective nodes and routers. The routers and the Domain Name Service (DNS) must also be set up correctly.

The transition from current IPv4 nodes to the IPv6 world will most probably follow a dual stack strategy. It is also foreseen however that new devices might be introduced on the network as IPv6 only nodes. Besides upgrading nodes and routers to IPv6 a few other issues need to be addressed for IPv6. This primarily affects the Domain Name Service DNS.

Applications are not supposed to be addressed directly via IP addresses but should use logical names. 32-bit addresses were used frequently in IPv4. However, this is too awkward and error-prone in IPv6, because the addresses are four times as long. The Domain Name Service, which is installed on a suitable server, is used to convert names to addresses and vice versa.

The BIND Version 9 Name Server defined by the Internet Software Consortium (ISC) is required for maintaining the IPv6 resource records (A6). If there is no IPv6-enabled DNS server in a network, the information on name and address conversion can also be maintained locally on the respective nodes. A host file is used for example in BS2000/OSD.

5.1 Basic transition mechanisms

RFC 1933 defines two basic mechanisms for converting from IPv4 to IPv6.

- Dual stack nodes and routers

  This mechanism fully supports both the IPv4 and the IPv6 protocol.

- IPv6 via IPv4 tunneling

  IPv6 packets are encapsulated in IPv4 packets and transported via an existing IPv4 network.

5.1.1 Dual stack

Dual stack nodes and routers can interoperate directly with both IPv4 and IPv6 partners. The dual stack nodes provide a user interface for this purpose, which enables access to both the IPv4 and the IPv6 network protocol. Name and address conversion functions, which are supported by the Domain Name Service, are an important component of these user interfaces.

5.1.2 Tunneling

IPv6 nodes or networks interlinked purely on the basis of IPv4 networks can be connected together by means of tunnels. IPv6 packets that pass through such a tunnel are encapsulated in IPv4 packets. The end points of the tunnel have both an IPv4 and an IPv6 address. There are two types of tunnel, configured tunnels and automatic tunnels. Configured tunnels are created using suitable configuration measures. The 6bone itself is probably the most familiar and biggest (in terms of size) example of a network containing mainly configured tunnels. Automatic tunnels on the other hand do not need manual configuration. The tunnel end-points are automatically determined by using IPv4 compatible IPv6 addresses.

5.2 The tools in system solutions

When introducing IPv6 in the internet and in corporate networks, one faces two different sets of problems:

– On the one hand, there are IPv6 islands isolated in the IPv4 world that want to communicate with one another.

– On the other hand, communication must be possible between the existing IPv4 world and the new IPv6 world.

The first problem can generally be solved by using dual stack routers and by tunneling IPv6 packets. The solution for the second problem is based on dual stack mechanisms, application gateways, NAT techniques or the temporary allocation of IPv4 address and IPv4 in IPv6 tunneling.

The mechanisms described here are based on tunnel technology and are designed to enable IPv6 communication between IPv6 islands isolated in the IPv4 world via the existing IPv4 infrastructure.

5.2.1 Configured tunnels

Manually configured tunnels can be used to connect individual IPv6 nodes or networks via the existing IPv4 infrastructure. Typically, configured tunnels are used between sites where traffic will be exchanged regularly.

Application area	**Sites**
IPv4 requirements	IPv4 interconnectivity between sites
IPv4 address requirements	≥ 1 per site
IPv6 requirements	None
IPv6 address requirements	Nothing special
Node requirements	IPv6 stack or dual stack
Router requirements	Dual stack
Other requirements	None

5.2.2 Automatic tunnels

Automatic tunnels are used as configured tunnels to connect separated IPv6 nodes or networks via the existing IPv4 infrastructure. Automatic tunnels are created dynamically when needed and broken up when no longer necessary. Typically, automatic tunnels are used between sites where the need for traffic exchange is only incidental. A prerequisite for the use of automatic tunnels is the use of IPv4 compatible IPv6 addresses by the communicating nodes. These addresses allow the nodes to derive the IPv4 addresses of the tunnel endpoints automatically.

Application area	**Sites**
IPv4 requirements	IPv4 interconnectivity between sites
IPv4 address requirements	≥ 1 per site
IPv6 requirements	None
IPv6 address requirements	IPv4-compatible IPv6 address
Node requirements	Dual stack
Router requirements	None
Other requirements	None

5.2.3 Tunnel brokers

Configured tunnels usually require cooperation of the two parties to set up the correct tunnel end points. The tunnel broker model is a concept to collect the necessary information for the nodes to set up the configured tunnels. A tunnel broker can be viewed as an IPv6 ISP offering connectivity through IPv6 over IPv4 tunnels. Current implementations are web based tools that allows interactive setup of the configured tunnel.

By requesting a configured tunnel, the node gets assigned an IPv6 address out of the address space of the tunnel provider. The relevant DNS entries are updated automatically.

The created tunnel will provide IPv6 connectivity between the tunnel provider's IPv6 environment and the isolated node.

Application area	**Node**
IPv4 requirements	Nothing special
IPv4 address requirements	1
IPv6 requirements	None
IPv6 address requirements	None
Node requirements	Dual stack, web browser
Router requirements	None
Other requirements	Tunnel server

5.2.4 6to4 concept

The 6to4 concept is applicable for the interconnection of isolated IPv6 domains in an IPv4 world. The router at the edge of an IPv6 domain creates a tunnel to the other IPv6 domain. The IPv4 endpoints of the tunnel are identified and addressed by part of the prefix of the IPv6 domain. This prefix is made up of a unique 6to4 Top Level Aggregator plus a Next Level Aggregator, which comprises the IPv4 address of the border router of the IPv6 domain. Another interesting effect of 6to4 is that it automatically derives a 48-bit long IPv6 address prefix from an IPv4 address. With this mechanism, users can start to deploy IPv6 without ever having to acquire IPv6 addresses officially. It is also very valuable if a user's ISP does not offer IPv6 support for cost reasons.

Application area	**Sites**
IPv4 requirements	IPv4 connectivity between the sites
IPv4 address requirements	≥ 1 per site
IPv6 requirements	Globally unique 6to4 address prefix
IPv6 address requirements	None
Node requirements	IPv6 stack
Router requirements	Implementation of special regulations for 6to4 forwarding
Other requirements	Creation of special DNS resource records for 6to4-address prefixes and the IPv4 addresses used as Next Level Aggregators

5.2.5 6over4 concept

6over4 interconnects isolated IPv6 nodes in a site through IPv6 in IPv4 encapsulation without explicit tunnels. A virtual link is created using a local IPv4 multicast group. IPv6 multicast addresses are mapped to IPv4 multicast addresses to be able to do IPv6 Neighbor Discovery. To route between the IPv6 internet and the 6over4 domain in an organization, at least one site router needs to be handle 6over4 routing.

Application area	**Nodes**
IPv4 requirements	IPv4 connectivity between the nodes
IPv4 address requirements	1 per node
IPv6 requirements	None
IPv6 address requirements	None
Node requirements	Dual stack
Router requirements	6over4 routing must be installed
Other requirements	To allow IPv6 nodes on different links to communicate with each other, IPv4 multicast packets must be forwarded by the routers.

5.2.6 Communication between IPv4 and IPv6 nodes

When IPv6 islands are installed and connected together using one or several of the above mechanisms, communication between the individual IPv6 nodes is enabled. Communication between the existing IPv4 node and the new IPv6 node is likewise important to establish. This can be done in several ways, either using an application gateway, via protocol translation at the network layer or by temporarily allocating an IPv4 address to the IPv6 node.

Note on protocol translators:

Typically, a protocol translator maps the fields in the header of one protocol to semantically similar fields in the header of another protocol. Policies are defined for the translation between IPv4 and IPv6 in RFC 2765 (SIIT) (see page 56). However, it is not uncommon in IPv4 for the application to have precise knowledge of information from the network layer. An example of this is FTP. This makes it necessary not only to translate the network layer packets from IPv4 and IPv6 but also to translate at the application layer, which however prevents use of the IPv6 security functions for these applications.

5.2.6.1 Dual stack

In the dual stack model, both IPv4 and IPv6 are implemented (dual-stacked) in all nodes and routers. That way, communication with IPv4 nodes takes place with the IPv4 protocol and communication with the IPv6 world takes place with the IPv6 protocol. The limitation of this approach is the need to allocate an IPv4 address as well as an IPv6 address for each new node. The IPv6 address space extension and simplified configuration cannot be exploited in this case.

Application area	**Site**
IPv4 requirements	IPv4 addressing plan and IPv4 routing plan
IPv4 address requirements	1 per node. Several per router
IPv6 requirements	IPv6 addressing plan and IPv6 routing plan
IPv6 address requirements	1 per node. Several per router
Node requirements	IPv4 and IPv6 stack
Router requirements	IPv4 and IPv6 stack
Other requirements	None

5.2.6.2 Limited dual stack

In the limited dual stack model, only the "server" is dual-stacked. New clients only have IPv6 addresses. A server node is defined as a node hosting enterprise internet services, such as DNS, web server, mail server etc. A client node is defined as a node not offering those services. With this approach, far fewer IPv4 addresses are used than in the dual stack model described above, but communication between new client nodes with IPv6 addresses and old clients with IPv4 addresses is no longer possible and can only be reenabled on the servers using special application layer gateways.

Application area	**Site**
IPv4 requirements	Use of existing IPv4 infrastructure
IPv4 address requirements	1 per server
IPv6 requirements	IPv6 addressing plan and IPv6 routing plan
IPv6 address requirements	1 per node. Several per router
Node requirements	IPv4 and IPv6 stacks on the severs, IPv6 stack on the clients
Router requirements	IPv4 and IPv6 stack
Other requirements	None

5.2.6.3 SOCKS64 concept

The SOCKS gateway is a gateway system defined in RFC 1928. It is an enhancement of the original SOCKS protocol, which allows IPv4 nodes to use a SOCKS gateway as a relay to the actual IPv6 destination node. The same principle can also be used for IPv6 nodes that want to connect to an IPv4 node. Use of the enhanced SOCKS protocol is particularly recommended if the existing IPv4 applications already used SOCKS. No changes are required in DNS to use the SOCKS protocol.

Application area	**Site**
IPv4 requirements	None
IPv4 address requirements	1 per node
IPv6 requirements	None
IPv6 address requirements	One or more per node
Node requirements	The applications must use the SOCKS protocol
Router requirements	None
Other requirements	Dual stack SOCKS server

5.2.6.4 SIIT protocol

The SIIT protocol describes a method to translate between IPv4 and IPv6. Translation is limited to the IPv4 or IPv6 packet header. The translator is operating in a stateless mode, which means that it has to edit every packet without any context knowledge of the earlier packets. This restriction means that IPv6 security mechanisms cannot be used with this translator. Furthermore, network protocol information that is transmitted at higher layers also cannot be translated. The assignment of a temporary IPv4 address to the IPv6 node is not part of the SIIT protocol.

Application area	**Site**
IPv4 requirements	None
IPv4 address requirements	1 temporary per node
IPv6 requirements	None
IPv6 address requirements	IPv4 mapped IPv6 address
Node requirements	IPv6 stack
Router requirements	None
Other requirements	None

5.2.6.5 NAT-PT concept

The NAT-PT concept defined in RFC 2766 allows direct communication between IPv6 only and IPv4 only nodes. Communication is based on use of a dedicated device that does the translation between IPv4 and IPv6 addresses and keeps state during each logical session. The NAT-PT device also includes an application layer gateway to make translation possible between IPv4 and IPv6 DNS requests and answers.

Application area	**Site**
IPv4 requirements	None
IPv4 address requirements	≥ 1 per site
IPv6 requirements	None
IPv6 address requirements	None
Node requirements	IPv6 stack
Router requirements	None
Other requirements	None

5.2.6.6 Bump in the Stack (BIS) concept

The bump-in-the-stack concept defined in RFC 2767 allows for the use of IPv4 applications on an IPv4 host to communicate with IPv6 only nodes. Added to the IPv4 stack are three modules that intervene between the application and the network layer:

1. an extension to the name resolver,
2. an IPv4/IPv6 address mapper,
3. a translator.

The main idea is that when an IPv4 application needs to communicate with an IPv6 only partner, the IPv6 address of the partner is mapped into an IPv4 address, which is only valid locally on the node. The IPv4 to IPv6 translation is performed according to SIIT.

One can view bump-in-the-stack as a special implementation of a NAT-PT within the IP stack of a node. Note that the technique can also be implemented as a library function on a dual-stack node without affecting the system kernel.

Application area	**Node**
IPv4 requirements	None
IPv4 address requirements	A private address area
IPv6 requirements	None
IPv6 address requirements	None
Node requirements	IPv4 and IPv6 stack with extensions
Router requirements	None
Other requirements	None

5.2.6.7 Dual Stack Transition Mechanism (DSTM)

The Dual Stack Transition Mechanism is a combination of two mechanisms:

– Assignment of IPv4 global addresses to IPv6 hosts (AIIH).
– Dynamic Tunneling Interface (DTI).

AIIH is based on cooperation between DNS and DHCPv6.

The main idea is that when a dual stack system wants to communicate with an IPv4 only node, it requests a temporary IPv4 address from the AIIH server for the duration of the communication. If an IPv4 only node wants to initiate communication, it first asks the DNS for an A resource record with the IPv4 address of the partner node. If this DNS server does not have a type A resource record, it will ask a DHCP server to allocate a temporary IPv4 address to the partner node. The DHCP server will send the assigned IPv4 address to the

DNS server so that it can respond to the request. At the same time, however, it sends a reconfigure command to the partner node so that this node can use the assigned IPv4 address as its own IPv4 address.

If the dual stack partner node is not assessable via an IPv4 infrastructure, it will encapsulate IPv4 packets in IPv6 packets and send them to a tunnel endpoint. The additional IPv6 header is stripped off here and the packet is routed via the IPv4 infrastructure. This encapsulation of IPv4 packets in IPv6 packets is performed by the Dynamic Tunneling Interface.

Application area	**Site**
IPv4 requirements	None
IPv4 address requirements	≥ 1 per site
IPv6 requirements	DHCPv6
IPv6 address requirements	None
Node requirements	IPv4 and IPv6 stack
Router requirements	None
Other requirements	None

5.3 Examples of typical conversion scenarios

The examples outlined in the next sections illustrate the transition from IPv4 to IPv6. The first more general examples highlight typical transition scenarios. The section “Examples from actual installations” on page 63 explains transition scenarios using real installations.

5.3.1 Large organizations with a lot of IPv4 addresses

This example looks at a large organization distributed over a number of sites. This organization has no shortage of globally unique IP addresses. For technical reasons, however, a complete transition of the organization is not possible at this time. The deployment of IPv6 will result in islands being created in the IPv4 world.

Possible transition mechanisms

Internal communication can be implemented on the basis of the following techniques:

– Dual stack
– Tunneling
– 6over4, is a network with multicasting is involved.

Translation is only necessary for IPv6 only nodes.

Connectivity with the outside IPv4 world requires no additional precautions. If the Internet Service Provider does not offer IPv6 connectivity, automatic or configured tunnels must be set up for communication with external IPv6 partners.

5.3.2 Large organizations with a few IPv4 addresses

This example shows the IPv6 transition for a large organization, which is distributed over a number of sites but does not have sufficient globally unique IPv4 addresses. As a result of this situation, private IPv4 addresses are being used in the intranet and a NAT is acting as the gateway to the internet. The IPv6 transition must also be completed gradually in this organization.

Possible transition mechanisms

The same situation applies in terms of internal communication as in the last example:

– Dual stack
– Tunneling
– 6over4, if a network with multicasting is involved.

Translation is only necessary for IPv6 only nodes.

However, the use of private IPv4 address space leads to distinct differences in terms of external communication.

If the Internet Service Provider does not offer IPv6 connectivity, automatic or configured tunnels must be set up on dual stack routers for communication with external IPv6 partners and must have at least one globally unique IPv4 address.

The following possibilities exist for communication with IPv4 partners:

– Use of the existing NAT

– Use of NAT-PT

– Use of DSTM to temporarily assign a node a globally unique IP address. If IPv6 only nodes are installed, a special transition mechanism is required.

5.3.3 Office with one IPv4 address

This example involves an office with a small number of nodes attached to a subnet. Communication with the outside world is performed on the basis of a NAT, which has an IPv4 address assigned to it.

Possible transition mechanisms

Internal communication	**External communication with IPv6 partners**
IPv6 IPv4 with private address space	1. IPv6 direct, if the ISP offers IPv6 functionality 2. Dynamic tunnel from the border routers to the partner nodes 3. Configured tunnel to systems operated by the ISPs that support IPv6 4. For communication with IPv4 partners: NAT-PT

5.3.4 New network

A completely new network with official IPv6 addresses is the subject of this example.

The Internet Service Provider selected should be able to offer IPv6 network access, because access to the external network can then take place via IPv6. Use of IPv4 addresses is not necessary.

If the selected ISP does not offer IPv6 network access, a dual stack border router should be installed. Furthermore, an IPv4 address and a configured or automatic tunnel are also necessary.

Possible transition mechanisms

An IPv6-enabled DNS server is installed for internal communication based exclusively on global IPv6 addresses.

If the ISP does not offer services such as dual stack DNS or NAT-PT for external communication with IPv6 only nodes, the following mechanisms are required for the site in addition to the IPv4 address:

– Dual stack DNS
– NAT-PT

No further expenditure is required for external communication if ISPs are used, who offer IPv6 only node services.

5.3.5 Internet Service Provider (ISP)

The primary function of an Internet Service Provider is to offer transit services between its customers, intranets, home networks, individuals and the rest of the internet. ISPs also offer a number of other services like Virtual Private Networks (VPN), DNS server, DHCP server, etc.

The details of switching to a different ISP are obviously impossible to specify in a document such as this one because of the variety of ISP configurations, services, topologies and administration/business constrains and relationships. It might be useful, however, to generally split ISPs into two categories.

– Backbone ISPs ISPs, who own a Top Level Aggregator (TLA) prefix
– Small ISPs ISPs, who do not own a Top Level Aggregator (TLA) prefix

Note that the above breakdown is in practice nested in that a small ISP may be a provider for an even smaller ISP in which case it will also need some of the mechanisms a backbone ISP may use etc.

Possible transition mechanisms

As a minimum ISPs, should offer the following services:

– Ensure IPv6 connectivity with the native IPv6 backbone.
  Native connectivity to the backbone is clearly most desirable here, but tunnelled (over IPv4) connectivity may satisfy initial demand.

– Ensure IPv6 connectivity for their customers.
  Direct IPv6 connectivity is also required here. Tunnelled connectivity may satisfy demand for a transition period in this case also.

Internally the ISP will most likely use native IPv6 and/or tunnels to interconnect its IPv6 enabled routers.

External communication has two important aspects in this example:

1. Connectivity to the IPv6 backbone and other ISPs

   This type of connectivity will depend on the actual size of the ISP. Large or backbone ISPs will almost certainly have connectivity to the IPv6 backbone and other ISPs. The details of the peering with other ISPs will be determined in mutual agreements in the same way that IPv4 peerings are.

   Smaller ISPs, of which there are significantly more than backbone ISPs, have to follow different logic. A small ISP will need to connect first to a larger ISP, which must not necessarily be a backbone ISP, in order to get IPv6 address space and connectivity to the IPv6 backbone. This gives smaller ISPs indirect access to the IPv6 backbone. The details for this are regulated in agreements. Obviously direct connectivity is desired but tunneled IPv6 connectivity may be perfectly adequate initially. The smaller ISP may then also make peering arrangements with other ISPs.

2. Connectivity with the ISP's customers

   The ISP has several options for passing on its IPv6 connectivity to customers and gaining a business benefit from this:

   – Direct IPv6 connectivity to the IPv6 backbone.
     Direct IPv6 connectivity is obviously desired but tunneled IPv6 connectivity may suffice initially.

   – A 6to4 gateway to the IPv6 backbone.

   – IPv6 connectivity per translation service, such as SIIT or NAT-PT.

   In practice a combination of the above is likely to be provided. The exact combination will depend on customer demand and the ISP's preferences. One thing that should always be kept in mind is that the end goal is for all the above mechanisms to be phased out as IPv6 gets more and more deployed. Therefore, the ISP should ensure that deployment of such mechanisms does not hinder native IPv6 deployment.

5.4 Examples from actual installations

Below are some examples for situations that might occur in real live environments. Please note that these examples are only meant as possible solutions and should not be considered the final or even the best solution for the given situation.

5.4.1 Isolated IPv6 node in an IPv4 domain

A corporate customer requires connection to a new bank system. An IPv6 node is installed in the customer's site for the IPv6 connection to the bank. The bank decided in favor of a IPv6 only installation to benefit from its enhanced functionality particularly, for example, in relation to security and authentication. The bank's border router (R2) is a dual stack router but the bank's network is IPv6 only.

Migration requirements

- The IPv6 node needs to communicate with all other nodes within the customer network.
- The use of translators is not permitted, in order to be able to use the encryption and authentication procedures of the IPv6 functionality.
- No changes should be made to the bank's network.

5.4.1.1 Suitability of transition mechanisms

IPv4 and IPv6 mechanisms allow the nodes within the customer installation to communicate with external or IPv6 nodes. Tunneling is required to send the IPv6 packets within IPv4 networks. Translators are not suitable because of the end-to-end connection required for security reasons. A dual stack must be installed on the customer's system. The border router (R1) must likewise be operated with IPv4 and IPv6, to enable routing within the IPv4 network.

The IPv4/IPv6 node should be configured to tunnel all IPv6 packets to the default IPv4/IPv6 router. The IPv6 packets are encapsulated in packets so that they can be sent to the router via the IPv4 infrastructure. The problem is how to configure the IPv6 address to allow neighboring solicitations to be transferred to the router based on configured tunneling. If you are configuring a tunnel to the router R1, you could tunnel through to the bank's router R2 and leave router R1 as a normal IPv4 router.

6over4 can be used if the network supports multicast routing. As 6over4 only works within the IPv4 domain, the router R1 must need to support IPv6 and also have a 6over4 interface configured. The node can then communicate with the router R1, by using IPv4 multicast packets. The rest of the communication path is handled by a different mechanism, such as configured tunneling or 6to4.

6to4 can be implemented in the R1 and R2 routers using their unique IPv4 addresses as a Next Level Aggregator ID to create a unique IPv6 address.

5.4.1.2 Solution 1

The IPv4 network supports multicasting in this solution. The following mechanisms are proposed here:

– Dual stack
– 6over4
– 6to4 or configured tunneling

Router

The 6over4 mechanism requires the border router R1 to be operated in the IPv4 network with an IPv6 and a 6over4 interface. The node and router do not need to be in the same segment, and in fact if they were there would be no requirement for 6over4. The router and the node are expected to have some IPv4 infrastructure between them. A configured tunnel must be set up between the R1 and R2 routers, so that IPv6 packets can be transmitted over the IPv4 internet. Routers R1 and R2 could use 6to4, but this would mean that router 2 would als have to support 6to4 and this would go against one of the requirements, namely that the bank's configuration should not be changed.

Node

The IPv4 node first needs to implement a dual stack, keeping its original IPv4 address and then implementing 6over4 to allow the node to use encapsulation of IPv6 packets within IPv4 multicast packets. Router R1 needs to be configured to support IPv6 routing. The node finds out its prefix by sending a router solicitation message encapsulated within an IPv4 multicast packet to router R1. The router will then return with a router advertisement message likewise encapsulated in an IPv4 multicast packet.

5.4.1.3 Solution 2

For this solution, the IPv4 network does not require multicast support. Dual stacks or configured tunnels should be used:

The node first needs to be implemented with the dual stack mechanism. If the automatic assignment of an IPv6 address should fail, a unique IPv6 address must be entered manually using the prefix of router R2. If this is not possible, the address can also be established by sending a router advertisement encapsulated in an IPv4 packet to router 2. A tunnel must then be configured manually from the node to the bank's router R2. Once this has happened, the node can begin communicating with the partner node.

Both solutions require the use of security functions. Whether MD5 is used as the authentication algorithm or DES-CBC as the encryption algorithm depends on what the bank's system uses.

Implementers should be aware that, in addition to possible security attacks against IPv6, security attacks against IPv4 must also be considered. Use of IP security at both IPv4 and IPv6 levels should nevertheless be avoided for efficiency reasons. For example, if IPv6 is running encrypted, encryption of IPv4 would be redundant except if IPv4 traffic analysis is felt to be a threat. If IPv6 is running authenticated, the authentication of IPv4 will add little.

Conversely, IPv4 security will not protect IPv6 traffic once it leaves the IPv6 over IPv4 domain. Therefore, implementing IPv6 security is required even if IPv4 security is available.

Although the above was written for secure transmission with 6over4, it is also particularly relevant to all security mechanisms used.

5.4.2 Small / medium organizations using a NAT

There are nine offices, each office is linked using a point-to-point connection. Each site has a DHCP, DNS and mail server. Routers are used between the offices to reduce traffic. Each router supports the RIP routing protocol. The network currently uses a private address space with a 192.168/16 prefix with each site using a /24 prefix.

Head office

The head offices in London has a DNS server and a NAT at the border to convert the non-globally unique IPv4 addresses to globally unique IPv4 addresses. This is used for security purposes as well as for enabling its own internal addressing structure. All external traffic and traffic that is destined for external sources is sent through the NAT. This external traffic is minimal with a large percentage of this being SMTP traffic. Additionally, head office has a firewall which is configured to route all incoming SMTP traffic from the ISP server's IP address to the internal mail router, which is a Linux machine running SENDMAIL. The internal mail router then looks at the domain name in the message header and sends it directly to the relevant mail server in each of the offices. The Proxy Daemon Squid and NAT run on the same machine. An intranet server runs on a separate Linux machine using Apache.

The NAT at head office is used for external communication and is assigned the IPv4 address 194.14.1.1. All external communication runs over this device.

Migration requirements

- To maintain a translator at the network border to maintain unrestricted online operation.
- To allow for communication with IPv4 only nodes at all times during the transition.
- Eventually eliminate all IPv4 traffic within the customer's network.

5.4.2.1 Suitability of transition mechanisms

IPv4 and IPv6 mechanisms

IPv4 and IPv6 mechanisms allow nodes to communicate with external IPv4 and IPv6 nodes within the customer installation.

Tunneling

Tunneling is required to send IPv6 packets within IPv4 networks.

Translators

A translator could be used to replace the NAT at the border of the network. All external communication is then handled by the translator. This also means that the internal structures of the IPv4 network remain unchanged with no deviation within the private network.

Dual stack with configured tunnels

Configured tunnels are required if the entire network infrastructure between the individual sites is not being converted to IPv6. While a complete transition of the network infrastructure to IPv6 would be easier in operational terms, this isn't always possible particularly in the area of the WAN outside the customer's control.

Dual stack with automatic tunnels

The use of automatic tunnels allows nodes to be updated to IPv6 before routers. However, an IPv4-compatible IPv6 address has to be assigned to each node for this purpose. Tunneling would then occur directly between end points. However, because the individual site networks are small and each site only has a few routers, it would be easier to upgrade the routers to dual stacks initially and dispense with automatic tunnels altogether.

Dual stack and NAT

A NAT is already used at the border of the network. All nodes in the organization can be upgraded to dual stack and the NAT then used to convert between IPv6 and IPv4 addresses. This would allow all nodes in the network to be able to communicate using IPv6 only with the translator being used to convert IPv6 headers to IPv4 headers. However, due to the use of private IPv4 addresses, direct communication with external IPv4 partner nodes is still not possible.

6over4

The internal network does not use multicasting so this mechanism is not relevant for this scenario.

Dual stack, 6to4 and translators

These mechanisms can be used to assigned a unique IPv6 address to the translator, by using the unique IPv4 address for the translator as well as a Next Level Aggregator. This wold give each internal node in an organization its own unique IPv6 address.

5.4.2.2 Solution 1

Private address space is currently used so that there is no problem with limited IPv4 addresses. The easiest way to approach the upgrade is to use dual stack on all nodes. This solution does not require the complex methods of encapsulation.

Mechanisms suggested in this solution include:

– Dual stack
– Translator
– 6to4 (optional)

Stage 1: Head office

Head office in London is tackled first, as most traffic will be routed through here. It is essential that this is the first to be upgraded to IPv6, so that communication with the regional sites will function properly. The order of implementation within head office is described in detail in the next sections.

Default router R1

The default router connects head office with all regional offices. A software upgrade is required to allow the default router to operate as a dual stack router. The router will treat IPv6 as an independent protocol so that RIPv6 will need to be activated and configured for IPv6.

DHCP server

Installation of a DHCP server depends on whether stateful autoconfiguration is to be used. In this case, the server has to be upgraded to dual stack:

– to allow allocation of IPv4 addresses from the private address space
– to accept stateful IPv6 addresses.

DNS server

The DNS server must be upgraded to IPv6.

Proxy server / Translator

The proxy server or translator must be bilingual. This is the point in the network where conversion between IPv6 and IPv4 takes place.The firewall does not need to be reconfigured assuming the data sent and received is in IPv4 format and the ISP does not migrate to IPv6.

Mail and file server

Once the above have been converted then the mail server and any file servers will need IPv6 installed. Again the mail server may require some extra configuration.

Workstation

Once all the upgrades described have been performed, the workstations will need to be upgraded to support IPv6. This can be in any order and there is no time limit.

Stage 2: Regional offices in London

Once the head office has been upgraded, the regional offices in London can be started. These can be carried out in whichever order desired. It is important, however, that the procedure followed in stage 1 be followed consistently:

1. Default router
2. DHCP server
3. DNS server
4. Other servers, e.g. mail servers
5. Workstation

Stage 3: Other regional offices

Once the London sites have been upgraded, the regional offices at other sites can be started. The order is as follows: Birmingham - Manchester - Glasgow - Edinburgh. This order does not have to be followed but it should be noted that if Glasgow is upgraded before Manchester and Birmingham, tunnels will have to be configured from Glasgow's router to the head office router. Implementation in each office should be followed in accordance with stage 1 and 2.

Stage 4: Final stage

Once all nodes within the organization have been upgraded to IPv6, the IPv4 component in each node can be deactivated to allow for just IPv6 traffic on the network. Deactivation will obviously have to be done in reverse order to the implementation i.e. disable IPv4 on the workstations first and routers last. The translator at the border will convert all IPv6 headers into IPv4 headers and vice versa. Normally it is difficult for translators to convert from IPv4 to IPv6, particularly when external to internal communication is initiated. Only one source in this case will be externally initiating communication and this will be the ISP server when sending SMTP traffic. The translator (will have to be an application translator) at the border of the network can detect and forward SMTP traffic to the correct node internally.

6to4 option

The 6to4 mechanism could be used in conjunction with the translator. The one unique IPv4 address associated with the translator could be assigned to the Next Level Aggregator field creating a globally unique IPv6 prefix. This would allow all nodes within the organization to have a globally unique IPv6 address and allow the NAT to receive either IPv6 or IPv4 packets.

5.4.2.3 Solution 2

If there was absolutely no need for the implementation of IPv6 within the organization or if all IPv4 applications required intensive configuration to convert for IPv6 support, there is another solution. In this case the NAT could just be upgraded to a translator supporting external IPv6 traffic and leaving the current internal infrastructure the same. This adds to the problem of how IPv4 nodes can work out how to send to an IPv6 address external to the organization. As all external traffic would be sent to the ISP address, the translator could be configured to send all external traffic to this one IPv6 address. The nodes could be configured manually to send any external data to a certain IPv4 address, which could be configured by routers to send on to the translator. This would be quite complex and it would be far easier, for long term administration and maintenance, to migrate the network to IPv6.

5.4.3 Introducing IPv6 in an ISP environment

The network of an Internet Service Provider consists of at least three main areas:

– The core network,
– The connections to other ISPs and
– The customer access network

The next two sections will discuss how an ISP can introduce IPv6 in these areas.

A couple of steps must be taken first for each area:

– Request IPv6 address space
– Register the IPv6 sites and the routing
– Set up DNS

5.4.3.1 Introducing IPv6 in the core network

It is not really necessary to introduce IPv6 into the core network. An ISP may decide to tunnel IPv6 over its existing IPv4 infrastructure. But if the ISP decides to introduce IPv6 into the core, this can be done in several ways.

An ISP might decide to install separate dual stack or IPv6-only routers in the core network. These will be interconnected by dedicated lines (ATM, PVCs, leased lines, etc.) or (if the routers are dual stack) by IPv6 in IPv4 tunnels over the existing IPv4 core infrastructure. Routing can be set up such that IPv4 packets are routed through the old IPv4 infrastructure and IPv6 packets are routed through the new IPv6 infrastructure. When dual stack routers are stable enough to be used in the core network, things become simpler. The ISP can configure the core routers as dual stack routers which will route both IPv4 and IPv6 packets.

Next a connection to the global IPv6 network should be made. This can be done by a direct IPv6 connection or by some tunneling mechanisms. If the core of the network supports IPv6 and the other ISP also supports IPv6, a direct link can be used to transport IPv6 packets. When there is no direct IPv6 connection, tunneling mechanisms must be used to reach the global IPv6 network. Automatic tunneling can be used for example (6to4). An ISP might decide to set up one or more routers at the edge of its network to act as 6to4 gateways. This enables other IPv6 islands to reach the ISP by 6to4 tunneling. An alternative to the use of dynamic tunnels is the use of static ones as is the case in the 6Bone.

5.4.3.2 Introducing IPv6 in the customer access network

The customer network consists of dial up and leased lines connected to an access router. There are two possibilities to introduce IPv6:

1. Upgrade existing access routers to dual stack routers. The dual stack routers connect both IPv4 and IPv6 routers.
2. Install separate IPv6 or dual stack routers. IPv4 customers connect to the old IPv4 access routers while IPv6 customers connect to the new access routers.

These IPv6 access routers must be connected to the global IPv6 network. If the core does not support IPv6, one of the transition mechanisms described in the section "The tools in system solutions" from page 50 onwards must be used. Automatic tunneling can be done with for example (6to4). An alternative to the use of dynamic tunnels is the use of statically configured ones. When the core network does support IPv6, the access routers can be connected to the nearest IPv6 core router (either by IPv4/IPv6 links or tunneling over IPv4).

When the customer is an IPv6-only site, the ISP might decide to provide some transition mechanisms to help the customer reach IPv4-only nodes. To do this the ISP can install a NAT-PT, for example.

5.4.4 Internet exchange points

Based on the address space distribution, we can distinguish two models for setting up an IPv6 internet exchange point (IE).

1. The more or less traditional model that is most common in the IPv4 world. In this case, each ISP connecting to the IE arranges its own IPv6 address space independently of the other ISPs. The prefix for this address space is exchanged between the ISPs.
2. An addressing model where the IE acts as an address space provider. In this case, the IE obtains a TLA and can assign NLAs from this TLA address space to connected ISPs. In order to obtain connectivity to the global internet, the IE operator needs to arrange for global transit through one or more global transit providers (TLA ISPs) which are connected to the IE. Implicitly, this means that the IE arranges transit for all connected ISPs that use the address space assigned to the IE. This requires quite a different business model for an IE operator than in model 1.

Models 1 and 2 described above require the following steps to be taken by the IE operator and/or the connected ISPs.

5.4.4.1 Model 1

The IE operator requests an NLA and receives globally unique address space. This can be an NLA from a transit provider (TLA provider) that offers connectivity for the IE infrastructure. A global prefix is preferred as next hop attribute in BGP4 (BGP4-IPV6).

- Addressing infrastructure on IE

  The router interfaces connected to the IE infrastructure are assigned addresses from the address space.

- Update the IPv6 registry

  The sites, allocations and route objects are registered.

- BGP announcements

  ISPs connected to the IE advertise their own address space, which is independent of that of the IE.

5.4.4.2 Model 2

The IE operator requests a (sub-) TLA from its regional registration organization. IE customers get the highest NLA from this (sub-) TLA.

- Addressing infrastructure on IE

  The router interfaces connected to the IE infrastructure are assigned addresses from the address space.

- Update the IPv6 registry

  The sites, allocations and route objects are registered.

- The IE operator assembles global transit ISPs (TLA ISPs)

  The IE should assemble several TLA ISPs that will provide connectivity to the global IPv6 network. Such TLA ISPs must agree to transit traffic from all customers connected to the IE.

- BGP announcements

  The transit providers for the IE address space announce the (sub-)TLA from the IE to the global IPv6 network. The IE customers are informed of all prefixes that can be reached by them and for which they have a contract with the IE operator. Customers (NLA ISPs) get a next level NLA from the IE operator. The NLA ISPs announce their NLA to the TLA ISPs. They also announce their NLA to other NLA ISPs of the IE if there is a peering agreement between them.

5.4.5 Avoiding using a NAT

Take, for instance, the case of two large, network-dependent organizations that must interface operations due to a merger and acquisition or a new business partnership. Suppose both of the enterprises have large IPv4-based networks that have grown from small beginnings. Both of the original enterprises have a substantial number of private IPv4 addresses that are not necessarily unique within the current global IPv4 address space. Combining these two non-unique address spaces could require costly renumbering and restructuring of routers, node addresses, domains, areas, exterior routing protocols and so on. This scenario is common in the current business climate, not only for mergers and acquisitions, but also for large outsourcing projects, where many nodes from the outsourcing partner must be integrated into one existing enterprise address structure. For these situations, IPv6 offers a convenient solution.

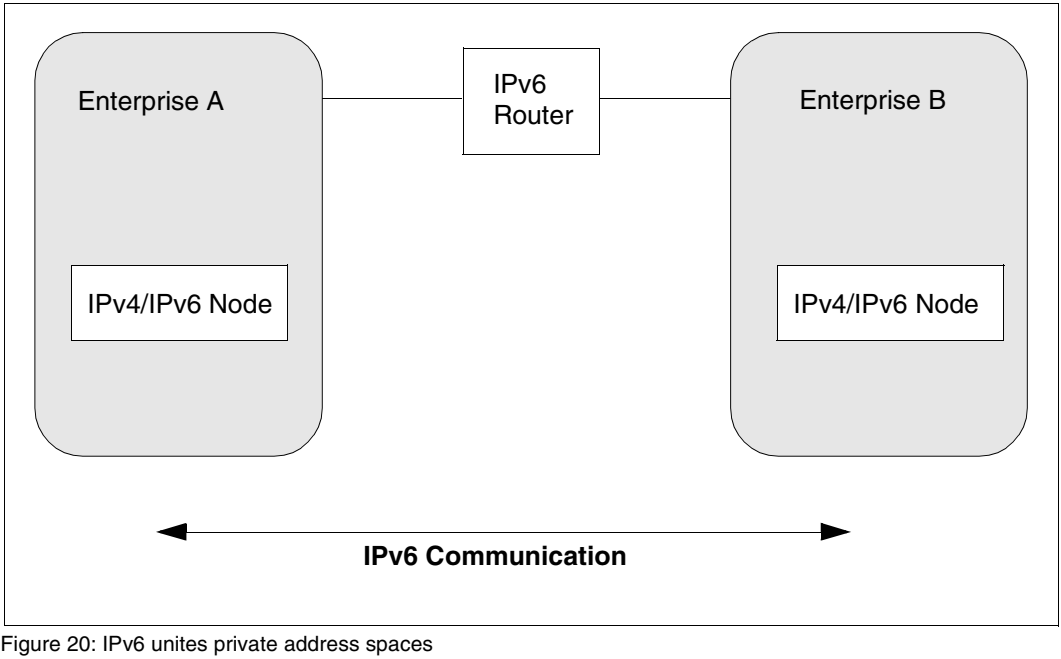


Figure 20: IPv6 unites private address spaces

The task of logically merging two enterprise networks into a single autonomous domain can be expensive and disruptive for the overall operation of the enterprise networks. To avoid the cost and disruption of comprehensive renumbering, enterprises may be tempted to opt for the stopgap solution of a network address translator (NAT). In the mergers and acquisitions scenario, a NAT could allow the two enterprises to maintain their private addresses more or less unchanged. To accomplish this, a NAT must conduct address translation in real time for all packets that move between the two organizations. Unfortunately, this

solution introduces all the problems associated with NATs that were already discussed, including performance bottlenecks, lack of scalability, lack of standards and lack of universal connectivity among all the nodes in the new enterprise and the internet.

In contrast with NAT, IPv6 seamlessly integrates the two physical networks (see Figure 18). Suppose the two originally independent enterprises are known as Enterprise A and Enterprise B. The first step is to determine which nodes need access to both sides of the new organization. These nodes are outfitted with dual IPv4/IPv6 stacks, which allow them to maintain connectivity to their original IPv4 network while also participating in a new IPv6 logical network that will be created with the assistance of the existing IPv4 physical infrastructure.

The accounting department of the combined enterprise will often have financial applications on servers that will need to be accessed by accounting employees in both Enterprise A and Enterprise B. Both servers and clients will run IPv6, but they will also retain their IPv4 stacks. The IPv6 sessions of the accounting department will traverse the existing local and remote links as "just another protocol," requiring no changes to the physical network. The only requirement for IPv6 connectivity is that routers that are adjacent to accounting department users must be upgraded to run IPv6. Where end-to-end IPv6 connectivity cannot be achieved, one of the IPv4/IPv6 tunneling techniques can be employed.

As integration continues, other departments in the newly merged enterprises will also be given IPv4/IPv6 hosts. As new departments and workgroups are added, they may be given dual-stack hosts, or in some cases, IPv6-only nodes. Nodes that require communication with the outside via the internet will likely receive dual stacks to maintain compatibility with IPv4 nodes exterior to the enterprise. But in some cases, hosts that only require access to internal servers and specific outside partners may be able to achieve connectivity with IPv6-only hosts. Migration to IPv6 presents the opportunity for a fresh start in terms of address allocation and routing protocol structures. IPv6 hosts and routers can immediately take advantage of IPv6 features such as stateless autoconfiguration, encryption, authentication and so on.

5.4.6 IPv6 from the edge to the core

For users, connectivity requirements typically focus primarily on access to local e-mail, WWW, database and applications servers. In this case, it may be best to initially upgrade only isolated workgroups and departments to IPv6, with backbone router upgrades implemented at a slower rate. IPv6 protocol development is more complete for internal routing than for high level backbone routing, so this is an excellent way for enterprises to gracefully move to IPv6. As shown in Figure 19, independent workgroups can upgrade their clients and servers to dual-stack IPv4/IPv6 nodes or IPv6-only nodes. This creates "islands" of IPv6 functionality.

As routing protocols, such as OSPF and BGP for IPv6 mature, the core backbone IPv6 connections can be deployed. After the first few IPv6 routers are correctly in place, it may be desirable to connect IPv6 islands together with router-to-router tunnels. In this case, one or more routers in each island would be configured as tunnel endpoints. As illustrated in figure 4 on page 22, when nodes use full IPv6 128-bit addressing, tunnels are manually configured so that the routers participating in tunnels know the address of the endpoints of the tunnel. With IPv4-compatible IPv6 addresses, automatic, nonconfigured tunneling is possible.

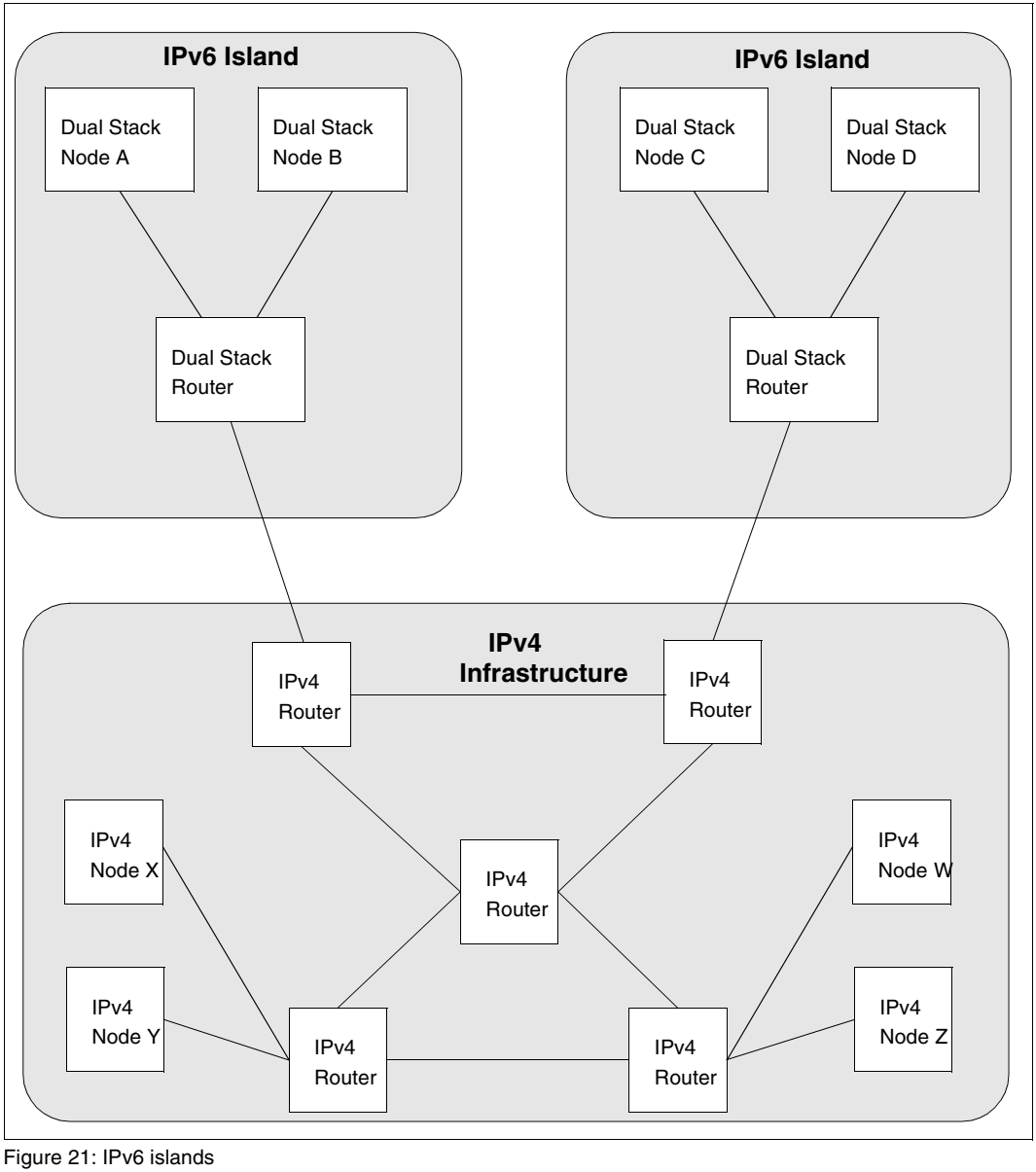


Figure 21: IPv6 islands

Routing protocols treat tunnels as a single IPv6 hop, even if the tunnel comprises many IPv4 hops across a number of different media. IPv6 routers running OSPF can propagate link-state reachability advertisements through tunnels, just as they would across conventional point-to-point links. In the IPv6 environment, OSPF can ensure that each tunnel is weighted properly within the topology. Routers generally make packet-forwarding decisions in the tunneling environment in the same way as in the IPv6-only network. The underlying IPv4 connections are essentially non visible to IPv6 routing protocols.

5.4.7 Other mechanisms

Additional mechanisms for transition or for IPv4/IPv6 coexistence are also under discussion. For example, IPv4 multicast can be used to support neighbor discovery by isolated IPv6 nodes. There are several proposals on how to support transactions between IPv4-only nodes and IPv6 nodes that do not have IPv4-compatible addresses.

IETF members are putting intense effort into transition, as well as the basic IPv6 protocol specification. The combination of tunnels, compatible addresses, dual-stack nodes and routers gives network administrators the range of flexibility and interoperability they need to deploy IPv6. Transition services allow organizations depending upon current IPv4 networks to take advantage of the more highly developed IPv6 features.

6 Using IPv6 in BS2000/OSD

This chapter looks at the specifics of BS2000/OSD in terms of the deployment of IPv6 in a network and provides concrete pointers for using the functionality implemented in IPv6 stage 1.

6.1 SOCKETS applications

The following sections present the changes required in the SOCKETS interface for use of IPv6. Also refer to the manual “SOCKETS(BS2000) V2.0” in this context.

6.1.1 Requirements

*open*Net Server Version 2.0 must be installed in order to used IPv6 as a network protocol and the SOCKETS application must be produced with SOCKETS(BS2000) V2.0. Furthermore, the SOC6 subsystem must be started.

6.1.2 Upgrading functions

socket()

When the *socket()* function is called, AF_INET6 must be entered instead of AF_INET as the *domain* parameter.

bind()

When the *bind()* function is called, a structure of type *sockaddr_in6* must be entered as a *name* parameter instead of a structure of type *sockaddr_in*. The *namelen* parameter must be set to the length of the *sockaddr_in6* structure.

connect()

When the *connect()* function is called, a structure of type *sockaddr_in6* must be specified as a *name* parameter instead of a structure of type *sockaddr_in*. The *namelen* parameter must be set to the length of the *sockaddr_in6* structure.

accept()

The *accept()* function returns a structure of type *sockaddr_in6* in the *addr* parameter instead of a structure of type *sockaddr_in*. The *addrlen* parameter is set to the length of the *sockaddr_in6* structure.

sendto()

When the *sendto()* function is called, a structure of type *sockaddr_in6* must be entered as a *to* parameter instead of a structure of type *sockaddr_in*. The *tolen* parameter must be set to the length of the *sockaddr_in6* structure.

recvfrom()

When the *recvfrom()* function is called, a structure of type *sockaddr_in6* must be entered as a *from* parameter instead of a structure of type *sockaddr_in*. The *fromlen* parameter must be set to the length of the *sockaddr_in6* structure.

gethostbyname()

The *gethostbyname()* function must be replaced by the *getipnodebyname()* function. The memory requested by the *getipnodebyname()* function must be freed again with the *freehostent()* function.

gethostbyaddr()

The *gethostbyaddr()* function must be replaced by the *getipnodebyaddr()* function. The memory requested by the *getipnodebyaddr()* function must be freed again with the *freehostent()* function.

getsockname()

The *getsockname()* function returns a structure of type *sockaddr_in6* in the *name* parameter instead of a structure of type *sockaddr_in*.

getpeername()

The *getpeername()* function returns a structure of type *sockaddr_in6* in the *name* parameter instead of a structure of type *sockaddr_in*.

inet_addr()

The *inet_addr()* function must be replaced by the *inet_pton()* function.

sinet_ntoa()

The *inet_ntoa()* function must be replaced by the *inet_ntop()* function.

htonl()

The *htonl()* macro for converting an IPv4 address from host format to network format is not required when using IPv6, because IPv6 addresses are always represented in network format.

ntohl()

The *ntohl()* macro for converting an IPv4 address from network format to host format is not required when using IPv6, because IPv6 addresses are always represented in network formatn.

Upgrading the address structure

The *sockaddr_in* structure must be replaced by the *sockaddr_in6* structure.

The following conversion rules apply for the individual structure elements:

`sin_family`	is replaced by	`sin6_familiy`	:= AF_INET6 instead of AF_INET
`sin_port`	is replaced by	`sin6_port`	:= port number as before
`sin_addr`	is replaced by	`sin6_addr`	:= IPv6 address instead of IPv4 address

The IPv6 address must be assigned with the *memcpy()* function. The INADDR6_ANY address must be used instead of the INADDR_ANY address.

6.1.3 Connectivity with IPv4-only partners

Communication with IPv4 only partners is no problem for IPv6 SOCKETS applications. Based on so-called mapped addresses, IPv4 addresses are treated in the same way syntactically as IPv6 addresses. The SOC6 subsystem recognizes with each function whether addresses are mapped and then executes the function as if it had been called in the AF_INET address family.

6.1.4 Examples

The next two examples show how SOCKETS applications can be upgraded to IPv6.

The modified or new code parts are highlighted in **bold**. If code parts are changed, the original IPv4-specific code part is shown below it as comment.

Example 1: Connection-oriented server for AF_INET6

```
#include <stdio.h>
#include <sys.types.h>
#include <sys.socket.h>
#include <netinet.in.h>
#include <netdb.h>

main(argc, argv)
int argc;
char *argv[];
{
#define TESTPORT 2222
int sock, length;
```

struct sockaddr_in6 server;

```
/* struct sockaddr_in server; */
```

struct in6_addr in6addr_any = IN6ADDR_ANY_INIT;

```
int msgsock;
char buf[1024];
int rval;
/* Create socket /
```

```
sock = socket(AF_INET6, SOCK_STREAM, 0);
/* sock = socket(AF_INET, SOCK_STREAM, 0); */

if (sock < 0)
{ perror("Create stream socket");
exit(1);
}

/* Assign a name to the socket */

server.sin_family = AF_INET6;
/* server.sin_family = AF_INET; */

memcpy(server.sin6_addr.s6_addr, in6addr_any,16) ;
/* server.sin_addr.s_addr = htonl(INDDR_ANY) ; */

server.sin6_port = htons(TESTPORT);
/* server.sin_port = htons(TESTPORT); */

if (bind(sock, &server, sizeof (server) ) < 0)
{ perror("Bind stream socket");
exit(1);
}
/* Start by accepting connection requirements */
listen(sock, 5);
msgsock = accept(sock, 0, (int *)0);
if (msgsock == -1)
{ perror("Accept connection");
exit(1);
}
else do {
memset(buf, 0, sizeof buf);
if ((rval = recv(msgsock, buf, 1024, 0)) < 0)
{ perror("Reading stream message");
exit(1);
}
else if (rval == 0 )
fprintf(stderr, "Ending connection\n");
else
fprintf(stdout, "->%s\n", buf);
} while (rval != 0);
soc_close(msgsock);
soc_close(sock);
}
```

Example 2: Connection-oriented client for AF_INET6

```
#include <stdio.h>
#include <sys.types.h>
#include <sys.socket.h>
#include <netinet.in.h>
#include <netdb.h>
#include <sys.uio.h>
main(argc, argv)
int argc;
char *argv[];
{
#define TESTPORT 2222
#define DATA "Here's the message ..."
int sock, length;
int error_num;

struct sockaddr_in6 client;
/* struct sockaddr_in6 client; */

struct hostent *hp;

struct hostent *getipnodebyname();


char buf[1024];

/* Create socket */

sock = socket(AF_INET6, SOCK_STREAM, 0);
/* sock = socket(AF_INET6, SOCK_STREAM, 0); */

if (sock < 0)
{ perror("Create stream socket");
exit(1);
}
/* Fill in address structure */

client.sin6_family = AF_INET6;
/* client.sin_family = AF_INET; */

client.sin6_port = htons(TESTPORT);
/* client.sin_port = htons(TESTPORT); */
```

```
hp = getipnodebyname(argv[1], AF_INET6, 0, &error_num);
if ((hp = = 0) || (error_num != NETDB_SUCCESS))
/* hp = getipnodebyname(argv[1], AF_INET6, 0, &error_num); */
/* if ((hp = = 0) || (error_num != NETDB_SUCCESS)) */

{ fprintf(stderr,"%s: unknown host\n", argv[1]);
exit(1);
}

memcpy((char *) &client.sin6_addr, (char *)hp->h_addr, hp->h_length);
/* memcpy((char *) &client.sin_addr, (char *)hp->h_addr, hp->h_length); */

/* Start connection */
if ( connect(sock, &client, sizeof(client) ) < 0 )
{ perror("Connect stream socket");
exit(1);
}

/* Write to the socket */
if ( send(sock, DATA, sizeof DATA, 0) < 0)
{ perror("Write on stream socket");
exit(1);
}
soc_close(sock);
}
```

6.2 DCAM and CMX applications

There are no changes for DCAM and CMX applications when IPv6 is used as the network protocol for communication with partner systems. However, it is necessary for the partner system to support the NEA or ISO transport service with IPv6 as the network protocol.

6.3 Administration

6.3.1 Configuration measures

Support for IPv6 and ICMPv6 is an option that can be enabled and disabled in BCAM with the BCOPTION IPV6=ON/OFF setting.

Only addresses complying with RFC 2373 are supported.
Since LAN addresses also find their way into these addresses via the interface identifier, we recommend you always define your own LAN addresses for local LAN/FDDI connections to avoid ambiguities.

Moreover, the same possibilities exist for defining IPv6 partner systems as in IPv4.

6.3.2 Autoconfiguration

The automatic generation of local IPv6 addresses is an option that can be enabled and disabled in BCAM with the BCOPTION IPV6-AUTO-CONFIG=ON/OFF setting.
If this option is enabled,

– *open*Net Server generates its own link-local IPv6 addresses

– *open*Net Server generates its own IPv4-compatible addresses

– *open*Net Server evaluates ICMPv6 prefix options sent by routers and generates appropriate addresses.

Moreover, the uniqueness of local IPv6 addresses is always checked.

6.3.3 Tunneling

Automatic tunneling is effective for IPv4-compatible IPv6 addresses.

Static IPv4 and IPv6 tunnels must be entered with MODIFIY-ROUTE ADD-IP-NET or ADD-IPV6-NET (see “BCAM” manual, Volume 1).

6.3.4 Configuration example

The following configuration example shows IPv6 systems being accessed locally, remotely and via IPv4 tunnels.

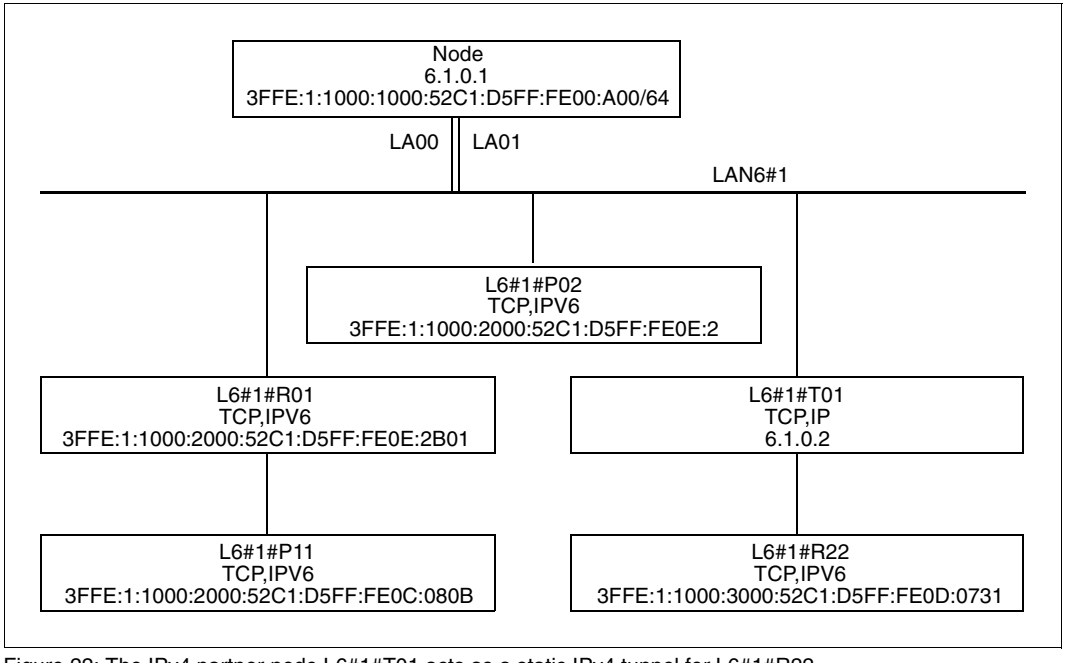


Figure 22: The IPv4 partner node L6#1#T01 acts as a static IPv4 tunnel for L6#1#R22

Static generation with KOGS macros:

```
*
*************************************************************************
* LAN **    LAN6#1                                                      *
*************************************************************************
*
  XLTNG UEPROZ=CSMACD,                                                  *
        LTGNAM=LAN6#1,                                                  *
        UEWEG=LAN,                                                      *
        DEVTYP=TRANSO,                                                  *
        DEVMN=(LAOO,LAO1),                                              *
        LANADR=5OC1D5OOOAOO,                                            *
        IPV6ADR=3FFE:1:1OOO:1OOO:52C1:D5FF:FEOO:AOO/64,                 *
        IPADR=OO6.OO1.OOO.OO1
*
  XKNOT KNOTNAM=LAN6#1
*
  XPRO  PROTYP=HOST/BCAM,                                               *
        PRONAM=L6#1#RO1,                                                *
        PROFIL=(TCP,IPV6),                                              *
        IPV6ADR=3FFE:1:1OOO:1OOO:52C1:D5FF:FEOE:2BO1,                   *
        NAKNO=YES
*
  XPRO  PROTYP=HOST/BCAM,                                               *
        PRONAM=L6#1#P11,                                                *
        PROFIL=(TCP,IPV6),                                              *
        IPV6ADR=3FFE:1:1OOO:2OOO:52C1:D5FF:FEOC:O8OB,                   *
        NAKNO=NO
*
  XPRO  PROTYP=HOST/BCAM,                                               *
        PRONAM=L6#1#PO2,                                                *
        PROFIL=(TCP,IPV6),                                              *
        IPV6ADR=3FFE:1:1OOO:1OOO:52C1:D5FF:FEOE:2,                      *
        NAKNO=YES
*
```

```
  XPRO  PROTYP=HOST/BCAM,                                          *
        PRONAM=L6#1#T01,                                           *
        PROFIL=(TCP,IP),                                           *
        IPADR=006.001.000.002,                                     *
        NAKNO=YES
*
  XPRO  PROTYP=HOST/BCAM,                                          *
        PRONAM=L6#1#R22,                                           *
        PROFIL=(TCP,IPV6),                                         *
        IPV6ADR=3FFE:1:1000:3000:52C1:D5FF:FE0D:0731,              *
        NAKNO=NO
*
  XEND
```

Dynamic generation with BCIN commands:

```
/BCIN LAN6#1,GEN=LOCAL,DEV=(LA00,LA01),                            -
/            PROFIL=(,(IPV6,IP),CSMACD),IPADR=(6,1,0,1),           -
/            I6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE00:A00/64', -
/            LANADR=X‘50C1D5000A00’

/BCIN L6#1#R01,GEN=NODE,ROUTE=LAN6#1,                              -
/              PROFIL=(TCP,IPV6),                                  -
/              I6-ADDRESS=‘3FFE:1:1000:1000:52C1:D5FF:FE0E:2B01’

/BCIN L6#1#P11,GEN=REMOTE,ROUTE=L6#1#R01,PROFIL=(TCP,IPV6),        -
/              I6-ADDRESS=‘3FFE:1:1000:2000:52C1:D5FF:FE0C:080B’

/BCIN L6#1#P02,GEN=NODE,ROUTE=LAN6#1,                              -
/              PROFIL=(TCP,IPV6),                                  -
/              I6-ADDRESS=‘3FFE:1:1000:1000:52C1:D5FF:FE0E:2’

/BCIN L6#1#T01,GEN=NODE,ROUTE=LAN6#1,PROFIL=(TCP,IP)               -
/              IPADR=(6,1,0,2)

/BCIN L6#1#R22,GEN=REMOTE,ROUTE=L6#1#T01,PROFIL=(TCP,IPV6),        -
/              I6-ADDRESS=‘3FFE:1:1000:3000:52C1:D5FF:FE0D:0731’
```

Dynamic generation with SDF commands:

```
/CREATE-NODE NODE-NAME=LAN6#1

/CREATE-LINE LINE-NAME=LAN6#1,-
/            IP-ADDRESS=6.1.0.1,-
/            IPV6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE00:A00/64', -
/            L2-PROTOCOL=*CSMACD(NODE-NAME=LAN6#1,                  -
/            WRITE-DEVICE=LA00,READ-DEVICE=LA01,                    -
/            LAN-ADDRESS=X'50C1D5000A00')


/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#R01

/CREATE-ROUTE ROUTE-NAME=L6#1#R01,                                  -
/       PATH=*NODE(NODE-NAME=LAN6#1,                                -
/       L3-PROTOCOL=*IPV6(                                          -
/       IPV6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE0E:2B01'))

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#P11

/CREATE-ROUTE ROUTE-NAME=L6#1#P11,                                  -
/             PATH=*VIA-ROUTER(ROUTER-ROUTE-NAME=L6#1#R01,          -
/             L3-PROTOCOL=*IPV6(                                    -
/             IPV6-ADDRESS='3FFE:1:1000:2000:52C1:D5FF:FE0C:080B'))

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#P02

/CREATE-ROUTE ROUTE-NAME=L6#1#P02,                                  -
/             PATH=*NODE(NODE-NAME=LAN6#1,                          -
/             L3-PROTOCOL=*IPV6(                                    -
/             IPV6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE0E:2'))

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#T01

/CREATE-ROUTE ROUTE-NAME=L6#1#T01,                                  -
/             PATH=*NODE(NODE-NAME=LAN6#1,                          -
/             L3-PROTOCOL=*IP(IP-ADDRESS=6.1.0.2))

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#R22

/CREATE-ROUTE ROUTE-NAME=L6#1#R22,                                  -
/PATH=*VIA-TUNNEL(TUNNEL-ROUTE-NAME=L6#1#T01,                       -
/L3-PROTOCOL=*IPV6(                                                 -
/IPV6-ADDRESS='3FFE:1:1000:3000:52C1:D5FF:FE0D:0731'))
```

Automatic node creation:

```
/BCOPTION AUTO-ES-CREATE=ON

/CREATE-NODE NODE-NAME=LAN6#1

/CREATE-LINE LINE-NAME=LAN6#1,                                          -
/            IP-ADDRESS=6.1.0.1,                                        -
/            IPV6-ADDRESS='3FFE:1:1000:1000:52C1:D5FF:FE00:A00/64', -
/            L2-PROTOCOL=*CSMACD(NODE-NAME=LAN6#1,                      -
/            WRITE-DEVICE=LA00,READ-DEVICE=LA01,                        -
/            LAN-ADDRESS=X‘50C1D5000A00’)

/CREATE-PROCESSOR PROCESSOR-NAME=L6#1#T01

/CREATE-ROUTE ROUTE-NAME=L6#1#T01,                                      -
/             PATH=*NODE(NODE-NAME=LAN6#1,                              -
/             L3-PROTOCOL=*IP(IP-ADDRESS=6.1.0.2))

/MODIFY-ADDRESS-ASSIGNMENT ROUTE-NAME=L6#1#T01,                         -
/               ADD-IPV6-NET=‘3FFE:1:1000:3000:/64’
```

Entry in the processor file:

```
L6#1#R01 IPV6 3FFE:1:1000:1000:52C1:D5FF:FE0E:2B01
L6#1#P11 IPV6 3FFE:1:1000:2000:52C1:D5FF:FE0C:080B
L6#1#P02 IPV6 3FFE:1:1000:1000:52C1:D5FF:FE0E:2
L6#1#R22 IPV6 3FFE:1:1000:3000:52C1:D5FF:FE0D:0731
```

Definition measures for routers

Local IPv6 addresses must generally be entered, for example for transmitting prefix information, router lifetime etc.

Definition measures for partner processors

Only local IPv6 addresses must generally be entered.

6.4 Restrictions

The next sections show which of the IPv6 features described in chapters 3 and 4 will not be supported in IPv6 stage 1 or will only be supported with restrictions.

6.4.1 Path MTU discovery

The path MTU discovery function is not implemented in the present version. However this does not imply any functional restriction in the operation of IPv6.

6.4.2 IPSEC

Because the IPSEC standardization process is not yet complete, IPSEC functionality is not supported in this version - IPv6 stage 1.

6.4.3 Hardware

When using HNC-91849, problems may arise with the parallel use of IPv6 and IPv4 multi-casting because of the limited number of multicast addresses that this device supports.

6.4.4 Quality of service

The Quality of Service standardization process is not yet complete, which is why the Quality of Service functionality is not yet supported in the version described here.

7 Appendix

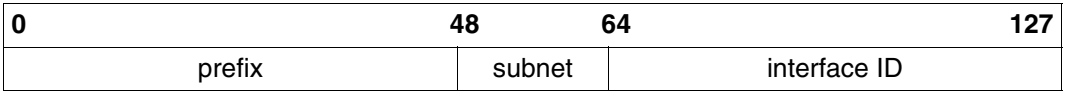
This appendix provides information on the following topics:

– IPv6 addressing rules
– IPv6 and DNS

7.1 Appendix 1: IPv6 addressing rules

7.1.1 IPv6 address assignment

Most of the transition mechanisms require dual stack systems and thus globally routable IPv6 addresses as well as globally routable IPv4 addresses. Although sometimes private IPv4 addresses [RFC1918] will suffice. But to allow communication between IPv4 and IPv6 nodes over the internet, at least one globally unique IPv4 address is always needed. Globally unique IPv4 addresses can be obtained from one of the Regional Internet Registries (IR), Local Internet Registries (LIR) or an Internet Service Provider (ISP). Without special registration, an IPv6 web site can deploy local web site addresses which are similar to IPv4 private addresses [RFC1918]. However, local web sites do not allow for communication over the internet. For this you need to apply for globally routable IPv6 addresses. Most sites will get a /48 prefix with 16 bits for subnetting and 64 bits for interface ID addressing. This means that 65536 subnets can be defined and in each subnet almost 20 trillion nodes can be numbered.



At present, there is an experimental network called the "6Bone" which is operated based on IPv6. A part of the address space is assigned for this network, the so-called Pseudo TLA (pTLA) 3ffe::/16. Provider pTLAs are assigned by the 6Bone. NLAs are assigned in turn by those organizations that have received pTLA assignments from the 6Bone.

Obtaining IPv6 address space

IPv6 addresses can be obtained from the same organizations as the ones who register IPv4 addresses. Basically, regional IRs delegate a part of the IPv6 address space to local IRs who further delegate parts of the address space to their customers. The smallest assignment that can be made to a customer is a /48 prefix. A difference between IPv4 and IPv6 allocations is that one of the main advantages of IPv6 allocation is route aggregation, i.e. to minimize the number of prefixes that need to be advertised in the default-free core of the internet.

The regional IRs use a slow start mechanism [IRALLOC] to allocate TLAs to ISPs. A special prequalification procedure [6PAPA] can be used by ISPs participating in the 6Bone.

ISPs can be divided into two categories: those ISPs that can get a (sub-)TLA from their regional Internet Registry (IR) and those ISPs that will not get a (sub-)TLA. In this document the first category is referred to as "TLA ISPs" and the second category is referred to as "NLA ISPs", because they will get an NLA from their upstream provider(s).

TLA ISPs will get a sub-TLA first and can apply later for a full /(29+n) prefix, where n ($0 \leq n \leq 19$) is the number of bits used to identify NLA ISPs [RFC 2374].

3	**13**	**13**	**19**	**16**	**64 bit**
FP	TLA ID	Sub-TLA	NLA ID	SLA ID	Interface ID
TLA ISP Prefix			NLA ISP Identifier (n bit)	Bit for NLA ISP	

An NLA ISP will be allocated a prefix between /29 and /48. It will use the remaining bits in the NLA ID to identify its customers. These customers will get a /48 prefix.

3	**13**	**13**	**19**	**16**	**64 bit**
FP	TLA ID	Sub-TLA	NLA ID	SLA ID	Interface ID
TLA ISP prefix			End customer site identifier (19 - n bit)	Bit for NLA ISP	

7.1.2 IPv6 registration issues

In the current IPv4 world, address space allocations are registered in the various databases managed by the regional IRs. Autonomous System (AS) information and routing policies are registered in the distributed Internet Routing Registry database (IRR). The IRs, LIRs and ISPs are supposed to register address space allocations and assignments, contact persons, AS numbers, routing policies and other useful data for network management in the various databases.

A special IPv6 registration database has been set up for the 6Bone community on the whois server named "whois.6bone.net". This is a special version of the RIPE database software and is referred to as the "6bone database". This database has special objects, the "inet6num:" object for assigned IPv6 prefixes and the "ipv6-site:" object which is used to register specific information about a site connected to the 6Bone, such as the configured tunnels and the origin AS. Objects can be found at the IPv6 site, namely IPv6 applications, which are supported at this specific site.

The database can be queried by using a modified whois client or the web-based "whois" service at http://www.6bone.net/whois.html. At this time, only the 6Bone database supports these special IPv6 objects. Currently, there are no database objects to register IPv6 routing policies.

When the regional IRs start allocating (sub-)TLAs the allocated and assigned IPv6 prefixes, routing policies etc. will have to be registered. At this moment it is unclear how exactly IPv6 registrations will be done.

Example of IPv6 address usage

Sites will get a /48 prefix. An example of how to use such a /48 is given below. In this example the site is allocated 3FFE:1234:5678::/48.

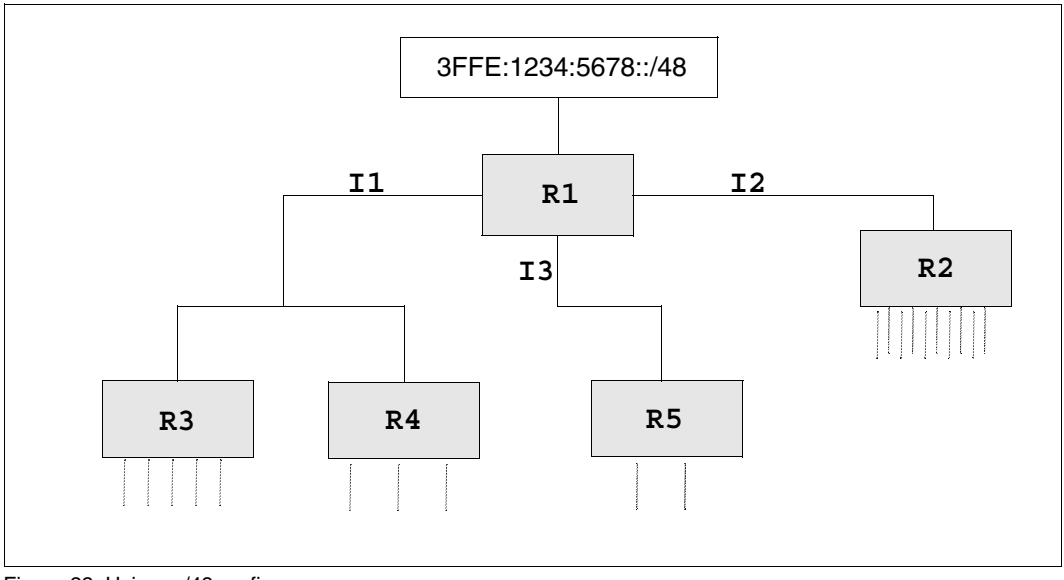


Figure 23: Using a /48 prefix

R[1-5] are routers and I[1-3] are the interfaces of R1. Suppose the expected number of nodes on the links is:

Router	Immediate	Year 1	Year 2
R2	34	50	70
R3	19	20	25
R4	9	10	15
R5	3	5	10

A number plan could be like the one shown in the table below. On R1 the following prefixes will be used on the interfaces:

```
I1        3FFE:1234:5678:2000::/50
I2        3FFE:1234:5678:0000::/49
I3        3FFE:1234:5678:2300::/50
```

Initially, R2 will get 256 /64s, R3 will get 48 /64s, R4 32 /64s and R5 16 /64s.

```
3FFE:1234:5678:0000::/50
------------------------
3FFE:1234:5678:0000::/49        I2
3FFE:1234:5678:1000::/49        free
3FFE:1234:5678:2000::/49        I1 + I3
3FFE:1234:5678:3000::/49        free
.....                           ...

3FFE:1234:5678:F000::/49        free
3FFE:1234:5678:0000::/49
------------------------

3FFE:1234:5678:0000::/64        interfaces of R2
.....                           ...
3FFE:1234:5678:00FF::/64        interfaces of R2
3FFE:1234:5678:0100::/64        reserved for R2
.....                           ...
3FFE:1234:5678:02FF::/64        reserved for R2
3FFE:1234:5678:0300::/64        free
.....                           ...

3FFE:1234:5678:2000::/49
------------------------

3FFE:1234:5678:2000::/50        I1
3FFE:1234:5678:2100::/50        reserved for I1
3FFE:1234:5678:2200::/50        reserved for I1
3FFE:1234:5678:2300::/50        I3
3FFE:1234:5678:2400::/50        reserved for I3
3FFE:1234:5678:2500::/50        reserved for I3
3FFE:1234:5678:2600::/50        free
.....                           ...
3FFE:1234:5678:2F00::/50        free

3FFE:1234:5678:2000::/50
------------------------

3FFE:1234:5678:2000::/64        interfaces of R3
.....                           ...
3FFE:1234:5678:202F::/64        interfaces of R3
3FFE:1234:5678:2030::/64        reserved for R3
.....                           ...
3FFE:1234:5678:204F::/64        reserved for R3

3FFE:1234:5678:2050::/64        interfaces of R4
.....                           ...
3FFE:1234:5678:206F::/64        interfaces of R4
```

```
3FFE:1234:5678:207O::/64        reserved for R4
.....                           ...
3FFE:1234:5678:209F::/64        reserved for R4
3FFE:1234:5678:20A0::/64        free
.....                           ...
3FFE:1234:5678:20FF::/64        free


3FFE:1234:5678:230O::/50
------------------------

3FFE:1234:5678:230O::/64        interfaces of R5

.....                           ...
3FFE:1234:5678:23OF::/64        interfaces of R5
3FFE:1234:5678:2310::/64        reserved for R5
.....                           ...
3FFE:1234:5678:231F::/64        reserved for R5
3FFE:1234:5678:232O::/64        free
.....                           ...
3FFE:1234:5678:23FF::/64        free
```

7.2 Appendix 2: IPv6 and DNS

7.2.1 Forward mapping

A node's 128-bit long IPv6 address can be stored in a Type AAAA record:

```
$ORIGIN ipv6.surfnet.nl.
...
zesbot          IN   AAAA   3FFE:0604:0000:0001:02C0:4FFF:FEC6:9CC7
```

This is comparable with the use of the Type A record in IPv4:

```
$ORIGIN ipv6.surfnet.nl.
...
zesbot          IN   A      192.87.110.60
```

Note that both A and AAAA records for a given zone are stored in the same DNS data file. A node with more than one IPv6 address must have more than one Type AAAA record:

```
$ORIGIN ipv6.surfnet.nl.
 ...
amsterdam9      IN   AAAA   3FFE:0600:8000:0000::0001
                IN   AAAA   3FFE:0600:8000:0000::0005
                IN   AAAA   3FFE:0600:8000:0000::0009
                IN   AAAA   3FFE:0600:8000:0000::000D
```

Currently a new record type, A6, is being defined to map a domain name to an IPv6 address, containing a reference to a "prefix" [DNSLOOKUP]. The aim of the A6 record is to facilitate network renumbering and multihoming. Domains employing the A6 record for IPv6 addresses can have automatically generated AAAA records to ease transition. After the A6 records are widely deployed it is expected that the AAAA records are no longer needed.

7.2.2 Reverse mapping

IPv4 uses the *in-addr.arpa* domain for reverse mapping. An IPv4 address is represented as a name in the *in-addr.arpa* domain by a sequence of bytes, written as decimal digits, separated by dots with the suffix *.in-addr.arpa*. The sequence of bytes is encoded in reverse order, i.e. the lowest order bytes are encoded first, followed by the next lowest order bytes and so on. For example, the IPv4 address 192.87.110.60 is represented as a name in the *in-addr.arpa* domain as:

```
60.110.87.192.in-addr.arpa.
```

This name is stored in a DNS data file as follows:

```
$ORIGIN 110.87.192.in-addr.arpa.
......
60              IN   PTR   zesbot.ipv6.surfnet.nl.
```

The special domain *ip6.int* is defined for IPv6 addresses. The process works exactly the same as with IPv4, except that an IPv6 address is represented by hexadecimal digits, separated by dots. For example the IPv6 address 3FFE:0604:0000:0001:02C0:4FFF:FEC6:9CC7 is entered in the *ip6.int* domain according to the rules as:

```
7.c.c.9.6.c.e.f.f.f.f.4.0.c.2.0.1.0.0.0.0.0.0.0.4.0.6.0.e.f.f.3.ip6.int.
```

This name is stored in a DNS data file as follows (assuming a /64 prefix):

```
$ORIGIN 1.0.0.0.0.0.0.0.4.0.6.0.e.f.f.3.ip6.int.
......
7.c.c.9.6.c.e.f.f.f.f.4.0.c.2.0   IN   PTR   zesbot.ipv6.surfnet.nl.
```

IPv4 and IPv6 reverse mappings are stored in different DNS data files.

Glossary

6Bone
Worldwide IPv6 test network.

additional header
Synonym for an extension header in the IPv6 protocol.

address autoconfiguration
IPv6 mechanism, which allows a node to automatically establish its network address and other information required for communication.

address resolution protocol
Protocol from the IPv4 environment for assigning network layer and link layer addresses.

anycast address
Special IPv6 address that can be used simultaneously by several nodes. The IPv6 network ensures, however, that an IPv6 datagram is only delivered to one node each time.

anycasting
Communication model based on the use of anycast addresses, which is especially suited for redundancy and failsafe concepts.

ARP
Abbreviation for address resolution protocol. See this entry.

authentication header
IPv6 extension header for authenticating the respective IPv6 packet.

automatic tunnel
Tunnel created automatically through use of IPv4-compatible IPv6 addresses.

backbone
A network whose task is to link several different interconnected networks, so-called frontend networks, on which systems and applications hang.

BGP

Abbreviation for border gateway protocol. See this entry.

BIND

Implementation of the Domain Name Service by the Internet Software Consortium.

border gateway protocol

Exterior gateway protocol.

bridge

Device or system that links two LANs and thus only operates on the data - essentially addresses - of the data link layer. A "normal" bridge, also referred to as a local bridge, links two LANs directly while a remote bridge links two LANs with the same technology via a WAN.

care-of address

Forwarding address for mobile nodes.

client

Term from the client/server architecture: the partner, who uses the services of the server.

configured tunnel

Tunnel created using manual configuration measures.

CSMA/CD

Carrier sense multiple access/collision detection. A procedure defined in IEEE 802.3 or ISO 8892-3 for LANs. Ethernet and 802.3 are similar, both work in accordance with the CSMA/CD procedure. The two terms are therefore often used synonymously.

datagram

Name for messages sent with connectionless communication. There is no guarantee that datagrams will arrive at the destination in the correct order or will not be duplicated.

destination options header

IPv6 extension header for the additional options of the respective IPv6 packet.

DHCP

Abbreviation for dynamic host configuration protocol. See this entry.

distance vector multicast routing protocol
Router protocol for setting up a distribution tree for multicast messages.

DNS
Abbreviation for domain name service. See this entry.

domain name service
System for administration of names and addresses on the internet.

draft
Preliminary version of an RFC.

DVMRP
Abbreviation for distance vector multicast routing protocol. See this entry.

dynamic host configuration protocol
Protocol for automatically configuring nodes on the internet.

encryption header
IPv6 extension header for encrypting the respective IPv6 packet.

extension header
IPv6 header containing optional information, which is not included in the IPv6 base header.

Ethernet
LAN introduced by XEROX, which is based on the Yellow Cable and uses CSMA/CD as its transmission method. Similar to an IEEE 802.3-LAN.

FDDI
Fiber distributed data interface: Procedure defined in ISO 9314 for LANs similar to the token ring but offering a higher speed.

file transfer protocol
Protocol for transmitting files on the internet.

firewall
Computer that protects a network against external attacks and other attempts to penetrate it from the outside.

forwarding address
Address at which a node is currently accessible.

fragmentation header
IPv6 extension header for fragmenting IPv6 packets into smaller packets.

FTP
Abbreviation for file transfer protocol. See this entry.

gateway
In general usage a system that links two or more networks and does not act as a bridge. Variants include gateways at the network layer (=router), transport layer gateways, application layer gateways.

hop by hop options header
IPv6 extension header for storing information that must be evaluated by every router, which forwards the respective IPv6 packet.

IAB
Abbreviation for internet architecture board. See this entry.

ICMP
Abbreviation for internet control message protocol. See this entry.

IETF
Abbreviation for internet engineering task force. See this entry.

internet
Communications architecture, characterized by the use of TCP and IP, which has developed from the ARPA network in the USA. Extensions are controlled by the IAB via the RFC process.

internet architecture board
Controls new developments on the internet based on the RFC mechanism.

internet control message protocol
Internet protocol used to transmit control and error information.

internet engineering task force
Controls new developments in the internet network based on the RFC mechanism.

internet service provider
Provider who provides access to the internet for its customers.

internet software consortium
A non-commercial organization, which performs standard implementations of important internet services. Supplier of the source codes of the DNS BIND server.

interworking
Generic term for linking all types of communication networks using different methods.

IPng
IP next generation, synonym for IPv6.

IPv4
Internet protocol: Connectionless network protocol of the internet architecture with 4-byte long addresses.

IPv6
Internet protocol: Connectionless network protocol of the internet architecture with 16-byte long addresses as a successor to IPv4.

IPv6 header
Header of the new IPv6 internet protocol.

ISC
Abbreviation for internet software consortium. See this entry.

ISP
Abbreviation for internet service provider. See this entry.

LAN
Local area network: Originally a high speed network with a low coverage. Today all networks, even wider coverage networks, which operate in accordance with CSMA/CD, token ring or FDDI.

link
Direct connection between two systems (node/router).

link layer
Communication link layer, which ensures direct communication between two systems (node/router) in a network.

message integrity code (MIC)
A code generated by an authentication procedure, which ensures the integrity of a message.

MOSPF
Abbreviation for multicast open shortest path first. See this entry.

MTU
Maximum size of a packet that can be transmitted over a network.

multicast address
Address used to send a message to several recipients in a network.

multicast open shortest path first
Router protocol for setting up a distribution tree for multicast messages.

multicasting
The sending of a message to several recipients in a network that are addressed by a group address.

NAT
Abbreviation for network translators. See this entry.

network layer
Communication network layer, which guarantees communication between different nodes in a network.

network translators
Network layer translator, which translates from a network protocol A to a network protocol B and vice versa.

next level aggregator
Second hierarchical level of an IPv6 address.

next level aggregator identifier
Identifier for the second hierarchical level of an IPv6 address.

NGTRANS
IETF working group that defines mechanisms for transition from IPv4 to IPv6.

NLA
Abbreviation for next level aggregator. See this entry.

NLA ID
Abbreviation for next level aggregator identifier. See this entry.

node

System, in which in contrast to an intermediate system, a transport entity (and thus also applications) reside. The node can physically comprise several hardware components.

open shortest path first

Interior gateway protocol.

OSPF

Abbreviation for open shortest path first. See this entry.

point to point protocol

Defined in RFC1171 and acts as a protocol between routers over serial lines for transporting or multiplexing different network protocols.

PPP

Abbreviation for point to point protocol. See this entry.

primary IPv6 header

First IPv6 header. Among other information, contains the address information of the IPv6 packet. This is the only IPv6 header ever available in an IPv6 packet.

renumbering

Renumbering of IP addresses of nodes and routers in a network or subnet.

RFC

Request for comment, an internet procedure for commenting on proposed standards, definitions or also reports, also a term for a document finalized using this method.

RIP

Abbreviation for routing information protocol. See this entry.

route aggregation

Assembly of several routes in routers to save storage capacity for routing tables and thus allow faster routing.

router

Element in a network that resides between networks and directs message streams through the networks, handling route selection, flow control, addressing and other functions in the process.

routing

The forwarding of messages through various networks.

routing header
IPv6 extension header for controlling the route of the respective IPv6 packet.

routing information protocol
Interior gateway protocol.

routing protocol
Protocol used by routers to exchange information with each other and with connected nodes in relation to topology, changes and costs of routes.

routing tables
Tables used by a router to forward incoming datagrams.

server
Logical entity or application component, which executes orders from clients and provides for the (coordinated) usage of generally available services (file, print, DB, communication,...), may also itself be a client for another server.

signature
A code generated by an authentication procedure, which guarantees the integrity of a message.

site level aggregator
Third hierarchical level of an IPv6 address.

site level aggregator identifier
Identifier for the third hierarchical level of an IPv6 address.

SLA
Abbreviation for site level aggregator. See this entry.

SLA ID
Abbreviation for site level aggregator identifier. See this entry.

TLA
Abbreviation for top level aggregator. See this entry.

TLA ID
Abbreviation for top level aggregator identifier. See this entry.

token ring
Technique used in token ring LANs, where a token traverses a ring-shaped LAN and regulates the right to send of the various stations.

top level aggregator
First hierarchical level of an IPv6 address.

top level aggregator identifier
Identifier for the first hierarchical level of an IPv6 address.

transport layer
Communication transport layer, guarantees communication between different applications residing on nodes in a network.

tunnel
Mechanism for encapsulating IPv6 packets in IPv4 packets for transmission via an existing IPv4 infrastructure.

unicast address
Address that can be used to send a message to precisely one recipient in a network.

WAN
Wide area network, public or private network that bridges large distances working relatively slowly - in contrast to LANs - with a higher error rate. These characteristics no longer apply in ATM networks.

Related publications

***open*Net Server V2.0** (BS2000/OSD)
BCAM V16.0A Volume 1
User Guide

Target group
The manual is intended for network planners, generators and administrators who use BCAM in BS2000 systems.
Contents
BCAM Volume 1 describes BCAM itself and how it is embedded in TRANSDATA and TCP/IP and ISO networks, plus generation and administrative activities. The description is clarified by generation examples and BCAM generation and diagnostics tools are described.

***open*Net Server V2.0** (BS2000/OSD)
BCAM V16.0A Volume 2
Reference Manual

Target group
The manual is intended for network operators, generators and administrators who use BCAM in BS2000 systems.
Contents
BCAM Volume 2 builds on Volume 1 and provides a detailed description of how to generate and operate the necessary BCAM commands. The KOGS macros required for static generation are presented and the BCAM error messages are listed.

***open*Net Server V2.0, *inter*Net Services V2.0** (BS2000/OSD)
SNMP-Management for *open*Net Server and *inter*Net Services
User Guide

Target group
The manual is intended for people responsible for networks and systems who want to use SNMP-based network and system management.
Contents
The manual provides a detailed description of the MIBs supplied with *open*Net Server, the FTP-MIB supplied with *inter*Net Services and the installation and operation of the sub-agents. A separate chapter contains an in-depth description of how to operate the BCAM Manager.

***inter*Net Services V2.0** (BS2000/OSD)
Administrator's Guide

Target group
This manual is intended for network planners, generators and administrators who want to use internet services in BS2000/OSD.
Contents
The manual describes the functionality of the internet services
BOOTP/DHCP, TFTP, DNS, FTP, LDAP, and NTP in BS2000/OSD. Installation, administration, operation, and logging and diagnostic options of the various components are also covered in the manual as well as FTP exits and TELNET exits.

***inter*Net Services V2.0** (BS2000/OSD)
User Guide

Target group
This manual is intended for network planners, generators and administrators as well as users who wish to use internet services in conjunction with BS2000/OSD.
Contents
The manual provides an introduction to the components of *inter*Net Services. It contains a detailed description of how to use FTP, the FTAC interface for FTP, and TELNET. Network administrators require this manual as a supplement to the Administrator's Guide.

***inter*Net Value Edition V1.0B** (BS2000/OSD)
User Guide

Target group
This manual is intended for network planners, generators and administrators who want to use the mail service in BS2000/OSD.
Contents
*inter*Net Value Edition is a supplement to *inter*Net Services that is available free of charge. The manual introduces users to the components of *inter*Net Value Edition and provides information on installation, administration and operation of the mail service in BS2000/OSD.

SOCKETS (BS2000) V2.0
SOCKETS for BS2000/OSD
User Guide

Target group
C programmers who want to develop communications applications in BS2000/OSD with the functions of the SOCKETS(BS2000) interface.
Contents
– Introduction to SOCKETS(BS2000)
– User functions in SOCKETS(BS2000)
– Installation and program generation

CMX (BS2000)
Communication Method in BS2000
User Guide

Target group
Programmers of transport service (TS) applications
Contents
CMX (BS2000) offers application programs a uniform interface to the transport services. You can use CMX (BS2000) to create application programs which can communicate with other applications independently of the transport system.

SNMP Management V5.0
SNMP Management for BS2000/OSD
User Guide

Target group
This manual addresses network administrators/operators and system administrators who wish to operate a BS2000 system or integrate it in SNMP-based management.
Contents
The manual describes how SBA-BS2, SSC-BS2, SSA-SM2-BS2 and SSA-OUTM-BS2 are embedded in BS2000/OSD, the installation and configuration procedures required to enable operation, and actual system operation. The agents and their MIBs which are required for monitoring are dealt with in detail. Installation and configuration of the relevant management applications on the management platforms Unicenter TNG, TransView SNMP and HP OpenView.
Other topics central to the manual include access to management information via the World Wide Web as well as the trap server for Solaris and Reliant UNIX.

IMON (BS2000/OSD)
Installation Monitor
User Guide

Target group
This manual is intended for systems support staff of the BS2000/OSD operating system.
Contents
The manual describes the installation and administration of BS2000 software using the IMON installation monitor and its three components IMON-BAS, IMON-GPN and IMON-SIC. Installation (standard and customer-specific) using the component IMON-BAS for systems with BS2000-OSD V2.0 and as of BS2000-OSD V3.0 is described in detail with the aid of examples in two separate chapters.

BS2000/OSD-BC
Introductory Guide to Systems Support
User Guide

Target group
This manual is addressed to BS2000/OSD systems support staff and operators.
Contents
The manual covers the following topics relating to the management and monitoring of the BS2000/OSD basic configuration: system initialization, parameter service, job and task control, memory/device/user/file/pubset management, assignment of privileges, accounting and operator functions.

Ordering RFCs

If the Requests for Comments (RFCs) referred to in the text are not included with delivery, they can be ordered in hardcopy form (copying charge) or fetched as a file from "anonymous internet FTP" or via e-mail.

Anonymous internet FTP: in order to fetch an RFC via the internet from the system *nic.ddn.mil* (IP address 192.67.67.20), please proceed as follows:
– set up an FTP connection to the system: *ftp nic.ddn.mil*
– you can now load the required documents from the directory *rfc*; a list of the available documents can be found in the file *rfc-index.txt*.

e-mail:

If you do not have internet access but can use electronic mail, you can request an RFC in this way. The document will be sent to you in response to your *Mail* query.

To do this, send a mail to the user *service* on the system *nic.ddn.mil*:
```
mail service@nic.ddn.mil
```
In the *Subject* field enter the number of the desired RFC, e.g.:
```
Subject: RFC 1155
```

Written queries concerning RFCs should be submitted to:

DDN Network Information Center
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025, U.S.A.
Phone: 415-859-3695

e-mail: nic@nic.ddn.mil

Please apply to your local office for ordering the manuals.

Index

Contents

*open*Net Server (BS2000/OSD)
IPv6 Introduction and Conversion Guide, Stage 1

User Guide

Target group
This manual is intended for everyone responsible for deciding as to the introduction of IPv6 in BS2000/OSD, as well as anyone using the IPv6 functionality on BS2000/OSD mainframes or planning to install IPv6 in BS2000/OSD.
Contents
The manual explains the commercial and technical foundations of IPv6. In addition, it describes the transition from IPv4 to IPv6 with the aid of examples and outlines the current status of the implementation of IPv6 in BS2000/OSD. Detailed information on “IPv6 addressing” and “DNS utilization” can be found in the appendix.

Edition: February 2001

File: ipv6_ums.pdf

Fujitsu Siemens computers GmbH
User Documentation
81730 Munich
Germany

Fax: (++49) 700 / 372 00000

e-mail: manuals@fujitsu-siemens.com
http://manuals.fujitsu-siemens.com

Comments
Suggestions
Corrections

Submitted by

Comments on *open*Net Server V2.0
IPv6 Introduction and Conversion Guide, Stage 1

Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format *...@ts.fujitsu.com*.

The Internet pages of Fujitsu Technology Solutions are available at *http://ts.fujitsu.com/*...
and the user documentation at *http://manuals.ts.fujitsu.com*.

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form *...@ts.fujitsu.com*.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter *http://de.ts.fujitsu.com/*..., und unter *http://manuals.ts.fujitsu.com* finden Sie die Benutzerdokumentation.