Chapter 2
IPv6 Networks

The mechanisms with which we exchange information have evolved significantly over the years. Despite its delays, the postal mail used to be the only method of communication a long time ago. Public switched telephony later made the process considerably faster and less laborious. More recently, the mobile cellular networks and IP networks allow information exchange with a much higher speed and reliability. All mechanisms that allow sending and receiving information require a supporting infrastructure that handles data delivery. For example, postal mails are sent from one geographical location to another through a series of exchange offices. The circuit-switched telephone networks also have dedicated network infrastructure. Similarly, an Internet Protocol (IP) network serves as infrastructure for most packet switched communication networks. Typically, an IP-based infrastructure comprises of hosts that send and receive information through a series of interconnected routers and gateways.

This chapter focuses on discussing the IP network in detail. This discussion is important because of two main reasons. Firstly, the idea of All-IP-Networks (AIN) is rapidly gaining popularity (Shin et al. 2013), which means that IP network shall operate as the backbone for all kinds of communication networks. Even the most recent technologies, like LTE, also provides means to connect to the IP network. Secondly, we are gradually shifting towards a new version of IP, namely IP version 6, which proposes several modifications in the legacy IPv4. Examining these modifications is important for understanding the operation of future IP networks. This chapter starts with a discussion on legacy IP-based networking infrastructure. After building the necessary foundation, this chapter discusses IPv6 in detail. The changes proposed by IPv6 in addressing and routing mechanisms have been examined in depth. The next two chapters of this book discuss innovative ways in which an IP network can be used and further enhanced. Chapter 2 examines the advantages of modifying the IP network by making it programmable while Chap. 3 explores the opportunistic use of the IP network.

S. F. Hasan, *Emerging Trends in Communication Networks*,
SpringerBriefs in Electrical and Computer Engineering,
DOI: 10.1007/978-3-319-07389-7_2, © The Author(s) 2014

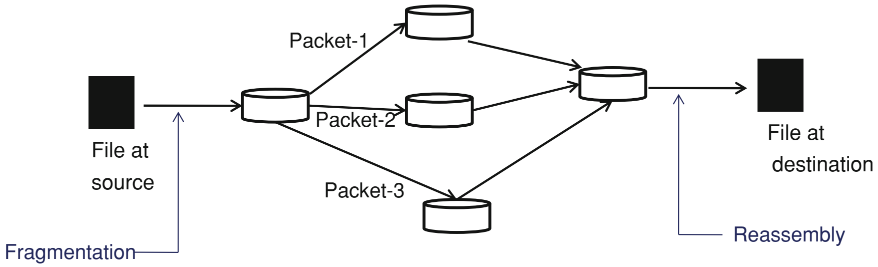


Fig. 2.1 A file is broken down into packets at the source (fragmentation) and put back together at the destination (reassembly). Each packet can traverse a different path

2.1 Basics of IP Networks

An IP network is a packet switched network that breaks the data to be sent into small packets. These packets are sent across a series of forwarding devices called routers. Routers receive packets at one interface, determine the next router using the so-called routing algorithms and send them out on an outgoing interface. Figure 2.1 shows the file transfer in a typical packet switched network. Note that packets originating from a file traverse through multiple routers before getting received at the destination via multiple paths. This method of data delivery is referred to as the datagram approach. In the datagram approach, a file to be sent is *fragmented* at the source into smaller packets. These packets are received at the destination and then *reassembled* back together in order. Fragmentation and reassembly have also been shown in Fig. 2.1.

Unlike circuit switching, packet switched IP networks do not reserve a dedicated path between source and destination. This feature of packet switching makes it suitable for supporting bursty traffic. A typical bursty traffic source does not transmit information consistently. Instead, there are some periods during which the amount of transmitted data is high whereas in other periods no (or very little) data is transmitted. Since most of the internet traffic is bursty in nature, packet-switched IP networks have become extremely popular. A simple packet capture on a live network can verify the bursty nature of internet traffic. Figure 2.2 shows internet traffic captured during a brief web browsing session. It is obvious that data transfer over the network is not consistent. If a dedicated path is reserved for a bursty source, as is the case in circuit switching, it shall remain unused during the periods when no data is transmitted. This results in under utilization of the network resources. Packet-switched networks address this issue by fragmenting a file into packets and deliver them to the destination over multiple hops.

Since multiple devices are involved in the data transfer process, it becomes necessary to uniquely identify the devices that handle data delivery. The identifiers used to name the devices in an IP network are referred to as IP addresses. The current infrastructure supports version 4 of IP (IPv4), in which an IP address is 4 bytes (32 bits) long. The hierarchy of an IP address is such that the first few bits identify the first hop router while the remaining bits of the address identify the device itself.

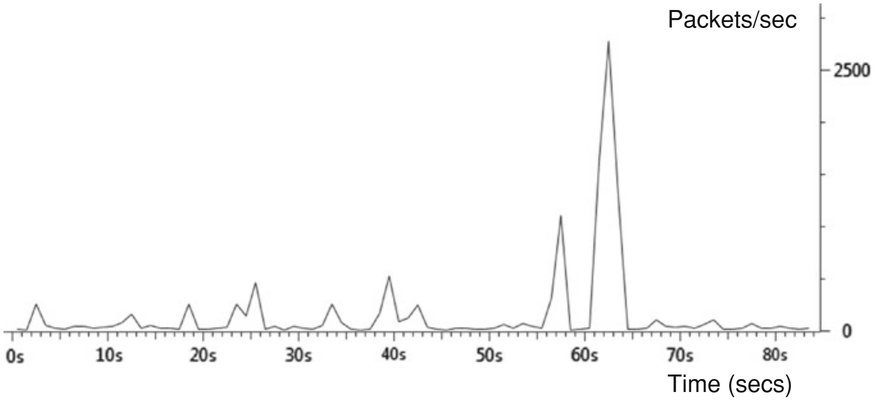
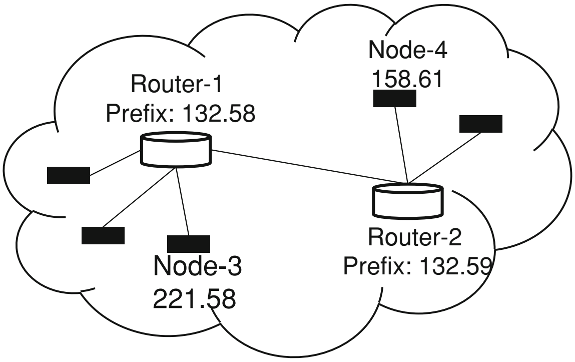


Fig. 2.2 Live packet capture shows the bursty nature of internet traffic

Fig. 2.3 Host and network parts in an IP address



Hence an IP address has a network part and a host part. The IP addresses are hierarchically similar to the telephone numbers which use the first few digits to identify the area and the rest of the digits to identify the phone itself. Figure 2.3 shows host and network parts of IP addresses such that the first 16 bits represent the network. The IP address of Node-3 in Fig. 2.3 is the combination of network and host parts, i.e. 132.58.221.58. All other devices connected to Router-1 have the same host part which is often referred to as the network prefix. The number of bits assigned to represent network prefix is known as the prefix length. In Fig. 2.3, all devices have prefix length of 16 bits. A node connected to a certain router uses its network prefix as the host part of the IP address. Node-4 in Fig. 2.3 uses the network prefix of Router-2, hence its IP address is 158.61.132.59.

The address space of an IPv4 network can support 2^{32} devices without address duplication. A few techniques, such as Network Address Translation (NAT), may be employed to accommodate more devices on the network (Wing 2010). However, the number of devices looking for a network connection has risen so much recently that a more aggressive addressing approach is required. By supporting 128-bit IP addresses, IPv6 significantly increases the number of devices that can connect to the network. The idea of increasing the address length has been around since 1996 (Li et al. 2007). However, recent increase in the number of network devices looking for network services has boosted the research and development activities in this area. In addition to increasing the address length, IPv6 networks also simplify the packet headers, making the routing process quicker and more efficient. The following sections first discuss the IPv6 address space and then highlight the modifications proposed by IPv6 in packet headers.

2.2 IPv6 Address

2.2.1 Format

An IPv6 address is a 128-bit case insensitive unsigned integer. It is represented as eight 16-bit hexadecimal words separated by colons. A typical example looks like:

$2001 : 0 : 5EF5 : 79FD : 24C8 : 85A : 8C6E : 6366.$

It is possible to represent an IPv6 address in other number systems, for example binary and decimal, etc. Hexadecimal representation is preferred because it is more compact and easy to handle. Multiple contiguous zero fields appearing in the IPv6 address can be merged together and written in a compressed form. The compressed form should appear only once in the address. For instance, the following IP addresses are the same:

$2001 : 0 : 5EF5 : 0 : 0 : 0 : 8C6E : 6366$ and $2001 : 0 : 5EF5 :: 8C6E : 6366$

where :: represent contiguous zero fields.
An unspecified address is an IPv6 address with all fields put to zero. This address is useful in the autoconfiguration process discussed later. Sample examples of an unspecified address are:

$0 : 0 : 0 : 0 : 0 : 0$ or equivalently ::

The complete representation of IPv6 address often requires its prefix length. An IPv6 address with prefix length of 64 bits is represented as:

$2001 : 0 : 5EF5 : 79FD : 24C8 : 85A : 8C6E : 6366/64.$

Such a representation indicates that the first 64 bits of the address identify the network while the rest identify the host. In IPv6 terminology, the network part of the address is referred to as the subnet prefix whereas the host part is known as the interface ID.

The interface ID is related to the device's MAC (or physical) address. An interface ID can be configured either manually or automatically. According to IEEE EUI-64, the interface ID is 64 bits long (Li et al. 2007) as shown in Fig. 2.4. Automatic

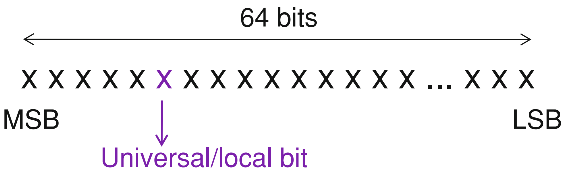


Fig. 2.4 Interface ID of IPv6 address according to IEEE EUI-64 format

configuration of interface ID requires (i) adding *FFEE* (in hexadecimal) in between the device's MAC address and (ii) inverting the universal/local bit. This converts 48 bit MAC address into 64 bit interface ID. The universal/local bit identifies whether the interface ID is defined locally or globally. The universal/local bit is set if an IPv6 address is locally defined.

The present day network devices often come with more than one network interfaces. For instance, a smart phone will have separate interfaces for 3G, Wi-Fi and WiMAX, etc. IPv6 can identify the device having multiple interfaces as well as the individual interfaces within the device. It uses Device User ID (DUID) to name a device that may have multiple interface cards. It also allows the use of Interface Association ID (IAID) in order to name each interface within the device.

2.2.2 Types of Addresses and IPv6 Messages

An IPv6 address is classified into three main types: Unicast, Multicast and Anycast address. A unicast IPv6 address identifies a single network interface to which packets are delivered. A multicast address, on the other hand, identifies a group of interfaces. The packets destined to a multicast address are received by all interfaces present in the group. It is also possible to send packets to a group such that they are received only by a single node. The receiving node may be selected by a routing algorithm based on how close it is to the source. This type of addressing is known as anycast addressing. Anycasting has two usage restrictions: (i) an anycast address must not be used as the source address, and (ii) anycast addresses are assigned to the routers.

IPv6 addresses can also be categorized as global and link-local addresses. Both kinds of addresses have different formats, as shown in Fig. 2.5 (Li et al. 2007). As the name indicates, a node uses global address to communicate over the network. This kind of an address is unique over the entire network. Link-local address in IPv6 is recognized by the prefix FE80 in hexadecimal. The link-local address is used in the Neighbor Discovery Protocols (NDP) for a variety of purposes (Alsa'deh and Meinel 2012). The Neighbor Discovery protocol helps nodes in detecting the available points of attachments. The protocol has five main purposes: router solicitation, router advertisement, redirection, neighbor solicitation and neighbor advertisement. All five NDP messages are carried in the Internet Control Messaging Protocol (ICMPv6) packets.

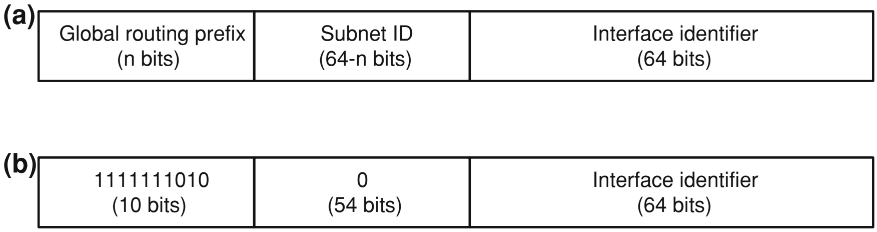

(a)
Global routing prefix (n bits)
Subnet ID (64-n bits)
Interface identifier (64 bits)
(b)
1111111010 (10 bits)
0 (54 bits)
Interface identifier (64 bits)

Fig. 2.5 Local and global IPv6 address

Router Solicitation messages are used by the nodes to query the information about the available routers. They are commonly used to identify a router when the device connects to the network. Routers often indicate their presence via the router advertisement messages. These messages are sent periodically and they contain the network prefix that is currently used by the router. The Router Advertisements used by NDP carry the link-layer address thus removing the need for resolving router's link-layer address. Router advertisements also carry the prefixes for every link, and allow address autoconfiguration. A link in IPv6 network can be associated with multiple prefixes. The information on all prefixes in use by the link are broadcast in the router advertisements (Narten et al. 2007).

The neighbor solicitation messages are used to query a neighboring node. It can be seen as a replacement of Address Resolution Protocol (ARP) request that verifies whether the neighboring node is active. Neighbor advertisement message is sent in response to the neighbor solicitation messages. The redirect messages are used by the routers to inform network devices about a possible alternate path to their respective destinations. These are particularly useful when a path faces failure or congestion.

2.2.3 Stateless Autoconfiguration

In IPv4 networks, devices dynamically acquire the IP address using the Dynamic Host Configuration Protocol (DHCP) (Blank 2002). This kind of address allocation is known as stateful autoconfiguration. IPv6 networks, on the other hand, allow the use of stateless autoconfiguration. In stateless autoconfiguration, DHCP server is not used and the address is generated by the node automatically.

The stateless autoconfiguration process proceeds as follows. A node first generates a link-local address as discussed in the previous section. This address is not assigned to an interface until its uniqueness is verified. In order to verify whether the link-local address is unique, Duplicate Address Detection (DAD) tests are performed. Several methods are available for performing DAD test (Thomson and Narten 2007). One simple method is that the node sends Neighbor Solicitation message with the address to be tested as the target address. If the address is already in use, the node already using the address shall send a response. If an address is not found to be unique, autoconfiguration shall not proceed further and the address shall have to be assigned using DHCPv6 server (or manually). On the other hand, if the link-local address is found to be unique, it is assigned to an interface. In the next step, the node listens to the Router Advertisement messages from the neighboring routers. As mentioned earlier, these messages will contain the network prefixes used by the advertising router. Using the information contained within the advertisement messages, the node generates an IPv6 global address. The uniqueness of this address may also be checked using DAD mechanisms. Once a node gets an IPv6 address, it can initiate DNS configuration. While IPv4 networks use dedicated DNS servers, DNS configuration is a little different in IPv6.

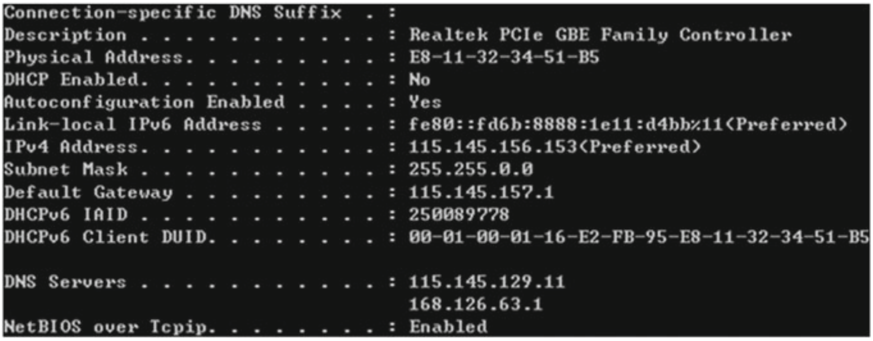
2.2.4 DNS Configuration

Domain Name System (DNS) servers are used in traditional IP networks to convert domain names into IP addresses. The information about the available DNS servers and their search list is contained in Router Advertisement and DHCP messages (Park et al. 2013). The recursive DNS server (RDNSS) and DNS search list (DNSSL) options have recently been introduced to carry DNS information. The RDNSS option contains the IPv6 address(es) of the recursive DNS servers that may be contacted for name resolution. On the other hand, DNSSL contains the domain names of DNS suffixes. In cases when RDNSS and DNSSL options are available from both sources Router Advertisement messages and DHCP, it is recommended to store at least three RDNSS addresses or DNSSL domain names (Jeong et al. 2010). Another possibility is to store the well known DNS server addresses in the IPv6 hosts's registry. This method is useful because, unlike other approaches, DNS configuration shall not require traffic exchange over the network. However, since the Internet Assigned Numbers Authority (IANA) have not assigned well known addresses to the DNS servers, this method cannot be used just yet. Nevertheless, (Park et al. 2013) consider this approach as a suitable candidate for DNS configuration in addition to other approaches.

This concludes our discussion on IPv6 addresses and their use in different operations. Most of the definitions covered so far can be seen in the IP configuration tool of the Disk Operating System (DOS). Figure 2.6 shows IP related information of a device that is connected to a network. The device is currently using IPv4 address because we have not switched to a standalone IPv6 network yet.

2.3 IPv6 Packet Headers

A packet in an IP network comprises of a header and a payload. The payload contains the actual data to be sent while the header contains information required for data delivery. In addition to increasing the address space, IPv6 also simplifies the packet



```
Connection-specific DNS Suffix  . :
Description . . . . . . . . . . . : Realtek PCIe GBE Family Controller
Physical Address. . . . . . . . . : E8-11-32-34-51-B5
DHCP Enabled. . . . . . . . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::fd6b:8888:1e11:d4bb%11(Preferred)
IPv4 Address. . . . . . . . . . . : 115.145.156.153(Preferred)
Subnet Mask . . . . . . . . . . . : 255.255.0.0
Default Gateway . . . . . . . . . : 115.145.157.1
DHCPv6 IAID . . . . . . . . . . . : 250089778
DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-16-E2-FB-95-E8-11-32-34-51-B5

DNS Servers . . . . . . . . . . . : 115.145.129.11
                                    168.126.63.1
NetBIOS over Tcpip. . . . . . . . : Enabled
```

Fig. 2.6 IP configuration tool showing IPv4 and IPv6 information

header. The information contained in the packet header is processed by routers to determine where a certain packet is to be sent. A large and complicated packet header would naturally require larger processing time. Various simplifications in packet headers have been proposed by IPv6 that increase the routing speed and allow quicker delivery of information. One of the major modifications in the packet header is the introduction of extension headers that are appended next to the base header. The base header only carries the necessary routing information while all supplementary information is available in the extension headers. A base header may have multiple extension headers. It is not necessary to use extension headers at all times. They are used whenever needed. The following discussion covers the base header and extension headers one by one.

2.3.1 Base IPv6 Header

The base IPv6 header is 40 bytes in length and contains fewer fields in comparison with the legacy IPv4 header (Brown and Parenti 2002). IPv6 header is the simplified version of IPv4 in that only necessary fields have been retained. Figure 2.7 shows the new IPv6 header with 128-bit source and destination addresses. As the name indicates, the version field in the base header specifies the version of IP (version 6 in this case) while the payload length indicates the size of data contained in the packet. The next header field specifies which extension header(s) follow the base header. If there are no extension headers, the next header field in the base header specifies the transport layer protocol that carries the IP packet. Common transport layer protocols include TCP and UDP etc. The priority field specifies the traffic class. This feature is used when the network faces congestion. If the congestion is severe, network has to drop a few packets to get things back to normal. Based on the priority level specified in the IPv6 header, network decides whether a particular header must be dropped during congestion. The packets containing real-time traffic generally have a higher

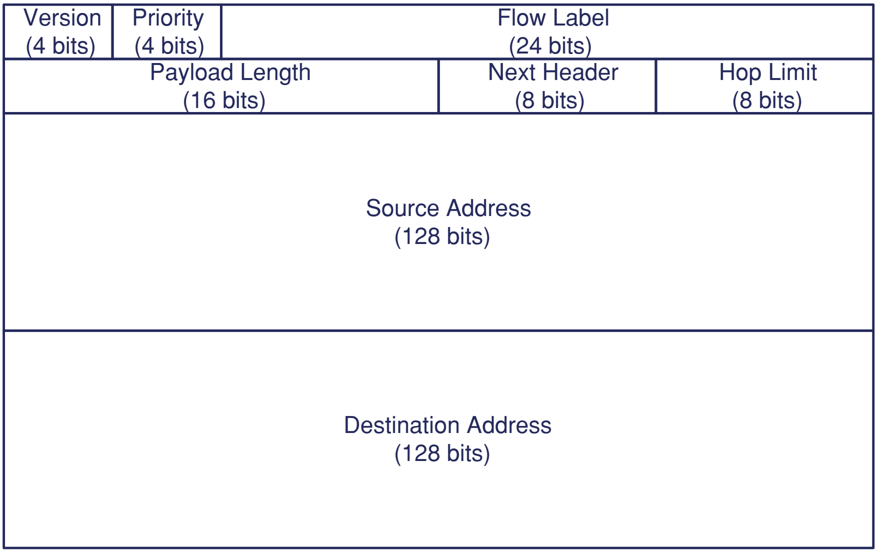


Fig. 2.7 IPv6 Header

priority and are seldom dropped during congestion. It is pertinent to mention that 3rd Generation Partnership Project (3GPP) has classified all applications into four categories: Background, Interactive, Streaming and Conversational. Each of these applications require a different quality of service and priority from the network. These priority requirements may be communicated to the network using the priority field. The hop limit field in the base header is used to identify and drop the packets that are stuck in indefinite loops. Upon receiving the packet, every intermediate router decreases the value specified in the hop limit by 1. A packet is dropped if its hop limit becomes 0 before it reaches its destination.

The flow label field in the IPv6 header specifies the packets belonging to the same *flow*. All packets in the same flow are treated in a similar manner by the intermediate routers. A router processes one packet from a flow and caches the results. These results are used to process the rest of the packets belonging to the same flow. This reduces the computational burden on the intermediate routers and speeds up the routing process. A flow may be defined in a variety of ways. For example, a flow may simply comprise of all packets originating from a certain source and destined to a certain destination as shown in Fig. 2.8.

2.3.2 Extension Headers

In legacy IPv4 networks, the IP header is followed by the transport layer header (TCP, UDP, etc). On the other hand, in IPv6 networks, there may be optional extension

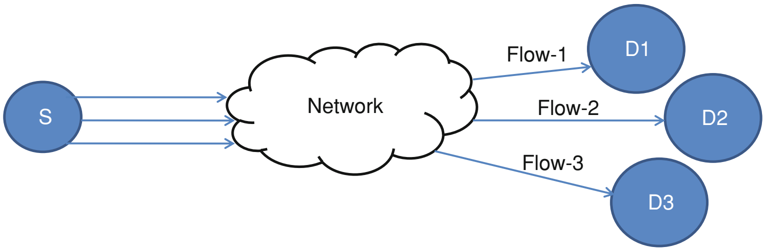


Fig. 2.8 Flow label distinguishes packets belonging to different flows

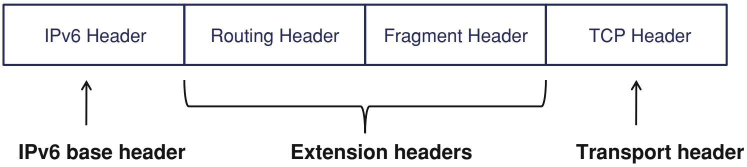


Fig. 2.9 Extension headers are placed between IPv6 base header and the upper layer transport header

Table 2.1 Extension headers specified for use in IPv6 (Loshin 2004)

Extension headers	Next header value
Hop-by-hop header	0
Routing header	43
Fragment header	44
Encapsulating security payload	50
Authentication header	51
No next header	59
Destination options header	60

header(s) between the IPv6 header and the transport layer header. If multiple extension headers are available, they are processed in the same order as specified by the source. Figure 2.9 shows a typical case where an IPv6 header is followed by two extension headers, which are then followed by the TCP header. Six extension headers have been specified for use in IPv6 networks. These have been tabulated in Table 2.1 (Loshin 2004). The value of the next header field to be used in the preceding header for each has also been given. Note that the table also contains an extension header called *No Next Header*. This is an imaginary header that implies that no extension headers are present. The use of such a header is also optional.

The hop-by-hop header appears immediately after IPv6 base header and is processed by all intermediate routers as well as the source and destination. On the other hand, the destination options header is processed only at the destination. The routing header is used at the source, which specifies the routers that are to be encountered by the packet. The number of intermediate routers is given in the Segment Left field of the header. The value contained in this field is decremented by 1 by every

Table 2.2 The fields used in routing header and their description

Fields	Description
Next header	Identifies the protocol header that follows
Header length	Specifies the length of the routing header
Routing type	Several types of routing headers have been specified
Segments left	Identifies the no. of route segments to be visited
Type specific data	Contains data according to routing type in use

intermediate router. As the name indicates, Fragmentation header is used for fragmenting and reassembling the packet. While a dedicated header for fragmentation is available, it must be noted that fragmentation is discouraged in IPv6 specifications. The Destination Options header contains variable length fields to enforce a few actions on the packet as it reaches the final destination. The Encapsulating Security payload and Authentication headers are meant to ensure security and privacy. These issues are out of this book's scope. A detailed discussion on IPv6 headers can be found in (Loshin 2004).

It is obvious from this discussion that different devices within the network are responsible for processing selected extension headers only. In IPv6, all devices do not have to process everything that comes their way. This makes processing headers less complicated than IPv4. In the following, we briefly discuss the routing header as an example of the extension headers. The purpose of this discussion is to show how extension headers are used in routing packets over the network.

The routing header contains the list of intermediate routers that a packet shall encounter as it moves from its source to the destination. These intermediate nodes are referred to as the route segments. The fields included in the routing header and their descriptions have been shown in Table 2.2. In addition to these fields, the routing header also contains source and destination addresses and the reserved field. The purpose of each of these fields has been explained in Fig. 2.10 (Li et al. 2007). The following explanation is for Type 0 routing headers. The figure shows that there are two route segments between source and destination. The source and destination addresses are changed every time the header reaches a route segment. As the packet reaches its first route segment, the segment left field is decremented by 1, and source and destination addresses are changed according to the next hop route segment. Note that the routing header contains the complete list of route segment addresses until it reaches its destination. At the destination, the segment field becomes 0 indicating that no more routing is required. The destination recognizes its address in the destination address field of the packet and accepts it as received.

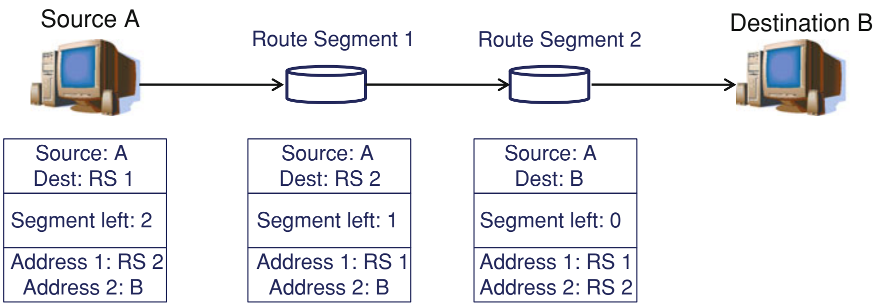


Fig. 2.10 An example to show the use of routing header

2.4 Migration to IPv6

Today's internet is built on IPv4 backbone. When the idea of IPv6 was introduced back in 1996, migration to IPv6 was considered only as a suggestion. Therefore, the process of migrating to an IPv6-only network has been very slow. There are three main players migrating from IPv4 to IPv6. The first is the service providers, who will migrate only when their service capacity is affected by the lack of IP addresses. The second are the enterprises, who will migrate only when their reachability and presence on the internet gets affected because of IPv4. Finally, there are end users, who simply do not care which address they are using (Kaur 2013). Both service providers and enterprises are realizing the need for migrating to IPv6. IPv6 was globally launched on 6 June 2012 with the help various participants including vendors, website and network operators (IPv6Launch 2012). The transition from IPv4 to IPv6 has begun and, more notably, has become almost mandatory due to the lack of IPv4 addresses.

Despite having sparse deployment of IPv6 around the world, much of the present IP infrastructure still supports IPv4. The complete migration to IPv6 will of course take time. The share of IPv4 in the current networking infrastructure is much larger, which is expected to decrease with the increasing deployment of IPv6 networks. Until this migration process is complete, IPv6 will have to co-exist with IPv4 infrastructure. There are three main strategies that have been adopted to allow this co-existence, as briefly highlighted in the following:

- Dual Stack: This approach allows a device to have two stacks, one each of IPv4 and IPv6 protocols. Dual stack is the easiest to implement and various operating systems have this feature built in already.
- Header Translation: This method uses algorithms to translate one header format into another. This is the least popular method which may also give rise to several complications.
- Tunneling: Tunneling is the most popular approach which allows IPv6 traffic to be encapsulated in IPv4 packets (and vice versa) when the traffic is passing through the IPv4 network. The intermediate routers shall treat all packets as v4. The IPv4 headers will be removed once the packets enter IPv6 domain.

2.5 Summary

This chapter describes the next generation of IP networks. IPv6 makes two main modifications in the legacy IPv4 protocol. Firstly, it increases the IP address length from 32 to 128 bits. This obviously increases the address space and hence the number of devices that can simultaneously access the network without address duplication. Secondly, IPv6 uses extension headers to simplify the base IP header. All extension headers used in packet delivery are not processed by all devices on the network. Respective devices process their concerned headers only, which speeds up the routing process. The current network, despite having larger portion of IPv4, is gradually migrating towards IPv6. In a few years time this migration shall become necessary, specially when techniques like machine-to-machine communications are implemented.

References

A. Alsa'deh, C. Meinel, Secure neighbor discovery: review, challenges, perspectives, and recommendations. IEEE Secur. Priv. **10**(4), 26–34 (2012)

A.G. Blank, TCP/IP Jumpstart: Internet protocol basics. (John Wiley and Sons, 2002)

E. Blanton, S. Chatterjee, S. Gangam, S. Kala, D. Sharma, S. Fahmy, P. Sharma. Design and implementation of the S3 monitor network measurement service on GENI. in 4th International Conference on Communication Systems and Networks (2012)

S. Brown, E. Parenti. Configuring IPv6 for cisco IOS. Syngress (2002)

Cisco. Software-defined networking: why we like it and how we are building on it. White Paper. (2013)

M. P. Fernandez . Comparing openflow controller paradigms scalability: Reactive and proactive. in International Conference on Advanced Information Networking and Applications (2013)

FloodLight. Project Floodlight: Open source software for building software-defined networks (2014) Available online at http://www.projectfloodlight.org/

GENI. Geni experiment and assignment repository (2013a) http://groups.geni.net/geni/wiki/GENIExperimenter/ExampleExperiments

GENI. Global environment for networking investigation. (2013b) Available online at geni.net.

S. Hares, R. White, Software-defined networks and the interface to the routing system (I2RS). Unknown Journal **17**(4), (2013)

Heller, Sherwood, McKeown. The controller placement problem. in ACM Hot SDN (2012)

IPv6Launch. (2012) http://www.worldipv6launch.org/. (available online)

M. Jarschel, F. Lehrieder, Z. Magyari, R. Pries.. A flexible openflow-controller benchmark. european workshop on software defined networking. (2012)

J. Jeong, S. Park, L. Beloeil, S. Madanapalli. IPv6 router advertisement options for DNS configuration. IETF RFC 6106 (2010)

G. Kaur. IPv6 Transition and deployment strategies–lessons from the trenches. LightReading webinar (2013)

Kurose, J. F., Ross, K. W., 2008. Computer networking: A top-down approach. IEEE Internet Comput. (Addison Wesley, 2008)

D. Levin, A. Wundsam, B. Heller, N. Handigol, A. Feldmann, *Logically Centralized?* State Distribution Trade-offs in Software Defined Networks,in *Workshop on Hot Topics in SDN, ACM Sigcomm* (2012)

Q. Li, T. Jinmei, K. Shima, *IPv6 Core Protocols Implementation* (Elsevier, Morgan Kauffman Series of Networking, 2007)

P. Loshin. IPv6: theory, protocol and practice. Morgan Kaufmann (2004)

N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner. Openflow: enabling innovation in campus network. in ACM SigComm (2008)

D. Meyer, The software-defined-networking research group. IEEE Internet Comput. **17**(6), 84–87 (2013)

Moy, J. T., OSPF: Anatomy of an Internet Routing Protocol (Addison-Wesley Professional, 1998)

T. Narten, E. Nordmark, W. Simpson, Neighbor discovery for IP version 6 (IPv6). IETF RFC 4861 (2007)

S. Ortiz, Software— defined networking: on the verge of a breakthrough? IEEE Comput. **46**(7), 10–12 (2013)

S. Park, J. Jeong, C.S. Hong, DNS configuration in IPv6: approaches, analysis, and deployment scenarios. IEEE Internet Comput. **17**(4), (2013)

T. Sedmak, Internet2 and networking industry partners drive innovation and development of SDN and openflow applications for 100G network. available online at internet2.edu (2013)

S.A. Shah, J. Faiz, M. Farooq, A. Shafi, S.A. Mehdi, An architectural evaluation of SDN controllers. IEEE International Communications Conference's Next Generation Networking Symposium **10**, 3504–3508 (2013)

D.H. Shin, D. Moses, M. Venkatachalam, S. Bagchi, Distributed mobility management for efficient video delivery over all-IP mobile networks: competing approaches. IEEE Network **27**(2), 28–33 (2013)

V. Thomas, GENI: Exploring networks of future.in *15th GENI Engineering Conference* (2012)

S. Thomson, T. Narten. IPv6 Stateless address autoconfiguration. IETF RFC 4862 (2007)

TraceRoute. Available online at. Network-Tools.com. (2013)

S.J. Vaughan-Nichols, Openflow: The next generation of the network. IEEE Computer **44**(8), 13–15 (2011)

D. Wing, Network address translation: extending the internet address space. IEEE Internet Comput. **14**(4), 66–70 (2010)

X. Yin, Huang, S. Wang, D. Wu, Y. Gao, X. Niu, M. Ren, H. Ma . Software defined virtualization platform based on double-flowVisors in multiple domain networks.in 8th International Conference on Communications and Networking in China (2013)

Springer

http://www.springer.com/978-3-319-07388-0

Emerging Trends in Communication Networks
Hasan, S.F.
2014, XVI, 76 p. 41 illus., Softcover
ISBN: 978-3-319-07388-0