3. Performance of IPv6 and Gigabit Networks

3.1 IPv6 Addressing Mechanisms

Chapter 2 introduced some basic concepts in IPv6 and their security ramifications. As we have seen, IPv6 (RFC 2460) is a connection less datagram protocol used for routing packets between hosts. However, there are a number of ancillary functions that support the main protocol and enhance overall performance. This chapter focuses on addressing and protocol structures. This chapter focuses on addressing and protocol structures related to high speed (Gigabit) networks in general and IPv6 in particular.

3.1.1 Introduction

From a packet-forwarding perspective IPv6 operates just like IPv4. An IPv6 packet, also known as an IPv6 datagram, consists of an IPv6 header and an IPv6 payload, as shown in Figure 3.1.1. The IPv6 header consists of two parts, the IPv6 base header, and optional extension headers. Functionally, the optional extension headers and upper-layer protocols, for example TCP, are considered part of the IPv6 payload. IPv4 headers and IPv6 headers are not directly interoperable: hosts or routers must use an implementation of both IPv4 and IPv6 in order to recognize and process both header formats; this gives rise to a number of complexities and security concerns in the migration process between the IPv4 and the IPv6 environments.

Fig 3.1.1 IPv6 Base Header.

The IPv6 addressing scheme is defined in RFC 3513 "The IPv6 Addressing Architecture specification" [9]. The IPv6 Addressing Architecture specification defines the address scope that can be used in an IPv6 implementation and the various configuration architecture guidelines for network designers of the IPv6 address space. Two advantages of IPv6 are that support for multicast is intrinsic (it is required by the specification) and nodes can create link-local addresses during initialization [RFC3315]. Some portions of this discussion are based on ideas proposed in [10].
3.1.2 Addressing Conventions

The IPv6 128-bit address is divided along 16-bit boundaries; each 16-bit block is then converted to a 4-digit hexadecimal number, separated by colons. The resulting representation is called *colon-hexadecimal*. This is in contrast to the 32-bit IPv4 address represented in dotted-decimal format, divided along 8-bit boundaries, and then converted to its decimal equivalent, separated by periods.

### IPv6 Address and Associated Reachability Scopes (Continued)

<table>
<thead>
<tr>
<th>Address Scope/Reachability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global addresses to reach the Internet (IPv6-enabled); also known as aggregatable global unicast addresses</td>
<td>Globally routable and reachable addresses on the IPv6 portion of the Internet (they are equivalent to public IPv4 addresses); global addresses are configured by router advertisement: . Global Unicast address . Other scope Multicast address Global addresses are designed to be aggregated or summarized to produce an efficient, hierarchical addressing and routing structure.</td>
</tr>
</tbody>
</table>

### IPv6 Address Space Allocation

<table>
<thead>
<tr>
<th>Address Space Allocation</th>
<th>Format Prefix</th>
<th>Percentage of the Address Space</th>
<th>Hex Notation</th>
<th>Fraction of the Address Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>0000 0000</td>
<td>0.391 %</td>
<td>Ox00</td>
<td>1/256</td>
</tr>
<tr>
<td>Reserved for NSAP allocation</td>
<td>0000 001</td>
<td>0.781 %</td>
<td>Ox0001</td>
<td>1/128</td>
</tr>
<tr>
<td>Aggregatable global unicast addresses</td>
<td>001</td>
<td>12.500%</td>
<td>001</td>
<td>1/8</td>
</tr>
<tr>
<td>Link-local unicast addresses</td>
<td>1111111010</td>
<td>0.098%</td>
<td>OxFE10</td>
<td>1/1024</td>
</tr>
<tr>
<td>Site-local unicast addresses (now deprecated)</td>
<td>1111111011</td>
<td>0.098%</td>
<td>OxFE11</td>
<td>1/1024</td>
</tr>
<tr>
<td>Multicast addresses</td>
<td>11111111</td>
<td>0.391 %</td>
<td>OxFF</td>
<td>1/256</td>
</tr>
<tr>
<td>The remainder of the IPv6 address</td>
<td>Unassigned</td>
<td>85.742%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The management of IPv6 site-local addresses is in many ways similar to the management of RFC 1918 addresses in some IPv4 networks. In theory, the private addresses defined in RFC 1918 should only be used locally, and should never appear in the Internet. In practice, these addresses leak. The conjunction of leaks and ambiguity ends up causing management problems. Names and literal addresses of private hosts leak in mail messages, Web pages, or files. Private addresses end up being used as...
source or destination of TCP requests or UDP messages, for example, in DNS or trace-route requests, causing the request to fail, or the response to arrive at unsuspecting hosts.

Having non ambiguous addresses will solve a large part of the router issues: since addresses are not ambiguous, routers will be able to use standard routing techniques, and will not need different routing tables for each interface. Some of the complexity will remain at border routers, which will need to filter packets from some ranges of source addresses; this is however a fairly common function.

Avoiding the explicit declaration of scope will remove the issues linked to the ambiguity of the site concept. Non-reachability can be obtained by using firewalls where appropriate. The firewall rules can explicitly accommodate various network configurations, by accepting or refusing traffic to and from ranges of the new non-ambiguous addresses.

### 3.1.3 Address Types

This section looks at some more detailed information related to address types. We discuss a number of unicast addresses, multicast addresses, and anycast addresses.

**Unicast IPv6 Addresses**

A unicast address identifies a single interface within the scope of the unicast address type. This could be a VoIP handset in a VoIPv6 environment, a PC on a LAN.

**Unspecified (Unicast) Address**

The unspecified address, 0:0:0:0:0:0:0:0 (that is, ::) indicates the absence of an address, and is typically used as a source address for POUs that are attempting to verify the uniqueness of a tentative address. It is equivalent to the IPv4 unspecified address of 0.0.0.0. The unspecified address is never assigned to an interface or used as a destination address.

**Loopback (Unicast) Address**

The loopback address, 0:0:0:0:0:0:0:1 or ::1, identifies a loopback interface, enabling a node to send POUs to itself. It is equivalent to the IPv4 loopback address of 127.0.0.1. POUs addressed to the loopback address are never sent on a link or forwarded by an IPv6 router.

**Compatibility (Unicast) Addresses**

IPv6 provides what are called 6to4 addresses to facilitate the coexistence of IPv4-to-IPv6 environments and the migration from the IPv4 to the IPv6 environment. The 6to4 address is used for communicating between two nodes operating both IPv4 stacks and IPv6 stacks (also known as dual stack) over an IPv4 routing infrastructure. The 6to4 address is formed by combining the prefix 2002::/16 with the 32 bits of the public IPv4 address of the node, forming a 48-bit prefix.

**Multicast IPv6 Addresses**

A useful feature supported in IPv6 is multicasting. The use of multicasting in IP networks is defined in RFC 1112 which describes addresses and host extensions for the way IP hosts support multicasting—the concepts originally developed for IPv4 also apply to IPv6. Besides a variety of protocol-level functionality supported by multicasting
(e.g., MLO and NO), one also can use this mechanism to support VoIP/IPTV functionality (e.g., audio-conferencing/bridging and program distribution). Multicast traffic is promulgated by utilizing a single destination address in the IPv6 header, but the IPv6 datagram is received and processed by multiple hosts. Hosts and devices listening on a specific multicast address comprise a multicast group; these devices receive and process traffic sent to the group address.

To identify all routers for the node-local and link-local scopes, the following multicast addresses are defined:

- FF01::2 (node-local scope all-routers address)
- FF02::2 (link-local scope all-routers address)

Next, we briefly look at solicited-node addresses. The solicited-node address supports efficient querying of network nodes for the purpose of address resolution. IPv6 uses the Neighbor Solicitation message to perform address resolution. This multicast address consists of the prefix FF02::1:FF00:0/104 along with the last 24 bits of the IPv6 address that is being resolved. In contrast to IPv4 where the ARP Request frame is sent via a Media Access Control (MAC)-level broadcast, and in doing so imposing on all nodes on the network segment, in IPv6 the solicited-node multicast address is used as the Neighbor Solicitation message destination. This avoids imposing on all IPv6 nodes on the local link by using the local-link scope all-nodes address.

**Anycast IPv6 Addresses**

An anycast address identifies multiple interfaces (typically belonging to different nodes), but not an entire broadcast universe. This could be used, for example, to support VoIP Voice Mail group distribution. With the appropriate routing topology, PDUs addressed to an anycast address are delivered to a single interface for further appropriate handling (a PDU addressed to an anycast address is delivered to the nearest interface identified by the address.) To make possible the delivery to the nearest anycast group member, the routing infrastructure must be aware of the interfaces that are assigned anycast addresses and must know their distances in terms of routing metrics. At present, anycast addresses are used only as destination addresses and are assigned only to routers. Note that an anycast address is syntactically indistinguishable from a unicast address. Thus, nodes sending packets to anycast addresses are not explicitly aware that an anycast address is being used.

### 3.1.4 Addresses for Hosts and Routers

In contrast to IPv4 where a host with a single network adapter has a single IPv4 address assigned to that adapter, an IPv6 host (e.g., a SIP proxy) typically has multiple IPv6 addresses (even in the case of a single interface). When a computer is configured with more than one IP address, it is referred to as a multi-homed system. IPv6 host and router address usage outlined in [10] is as follows:

**Host:** Typical IPv6 hosts are logically multi-homed because they have at least two addresses with which they can receive PDUs. Each host is assigned the following unicast address:
- A link-local address for each interface. This address is used for local traffic.
- An address for each interface. This could be one or more global addresses.
- The loopback address(::1) for the loopback interface.

Additionally, each host is listening for traffic on the following multicast addresses:
- The node-local scope all-nodes address(FF01::1)
- The link-local scope all-nodes address(FF02::1)
- The solicited-node address for each unicast address on each interface
- The multicast addresses of joined groups on each interface

**Router:** An IPv6 router is assigned the following unicast addresses:
- A link-local address for each interface. This address is used for local traffic.
- An address for each interface. This could be one or more global addresses.
- The loopback address (::1) for the loopback interface.

An IPv6 router is assigned the following anycast addresses:
- A subnet-router anycast address for each subnet
- Additional anycast addresses(optional)

Each router is listening for traffic on the following multicast addresses:
- The node-local scope all-nodes address(FF01::1)
- The node-local scope all-routers address(FF01::2)
- The link-local scope all-nodes address(FF02::1)
- The link-local scope all-routers address(FF02::2)
- The solicited-node address for each unicast address on each interface.
- The addresses of joined groups on each interface

### 3.2 More Advanced IPv6 Protocol Mechanisms

#### 3.2.1 Introduction

The previous two sections provided an introduction to IPv6. This section provides additional details and focuses on security-related issues.

As we have seen, like IPv4, IPv6 is a connectionless datagram protocol used primarily for addressing and routing packets between hosts. Connectionless means that a session is not established before exchanging data. Connectionless protocols are "unreliable" in the sense that delivery is not automatically guaranteed. IPv6 always makes a best-effort attempt to deliver a packet. An IPv6 packet might be lost, delivered out of sequence, duplicated, or delayed. IPv6 per se does not attempt to recover from these types of errors. The acknowledgment of packets delivered and the recovery of lost packets are done by a higher-layer protocol, such as Transmission Control Protocol (TCP). Other supportive protocols include the following: Internet Control Message Protocol for IPv6 (ICMPv6) (RFC 2463), Neighbor Discovery (ND) (RFC 2461), and Multicast Listener Discovery (MLD) (RFC 2710, RFC 3590, RFC 3810). ICMPv6 is a mechanism that enables hosts and routers that use IPv6 communication to report errors and send simple status messages. ND is a mechanism that is used to manage node-to-node communication on a link. ND uses a series of five ICMPv6 messages. ND replaces Address Resolution Protocol (ARP), ICMPv4 Router Discovery, and the ICMPv4 Redirect message; it also provides additional functions. ND is implemented using the Neighbor Discovery Protocol (NDP). MLD is a mechanism that enables one to manage subnet multicast membership for IPv6. It uses a series of three ICMPv6 messages and replaces the Internet Group Management Protocol (IGMP) v3 that is employed for IPv4.

#### 3.2.2 IPv6 and Related Protocols (Details)

We introduced a number of basic IPv6 concepts in previous sections. The sections that follow focus on a more formal description of IPv6. The discussion is based on IETF RFC 2460 [RFC2460]. There is an extensive body of technical research literature on this topic.
IPv6 is a new version of the Internet Protocol, designed as the successor to IP version 4 (IPv4) described in RFC 791. RFC 2460 specifies the basic IPv6 header and the initially defined IPv6 extension headers and options. It also discusses packet size issues, the semantics of flow labels and traffic classes, and the effects of IPv6 on upper-layer protocols. The format and semantics of IPv6 addresses are specified separately in RFC 2373 (now obsoleted by RFC 3513). The IPv6 version of ICMP, which all IPv6 implementations are required to include, is specified in ICMPv6 (RFC 2483). Normative guidelines for developers are provided in all relevant IETF RFCs.

Note:--------------------------------------------------------
It is possible, though unusual, for a device with multiple interfaces to be configured to forward non-self-destined packets arriving from some set (fewer than all) of its interfaces, and to discard non-self-destined packets arriving from its other interfaces. Such a device must obey the protocol requirements for routers when receiving packets from, and interacting with neighbors, the former (forwarding) interfaces. It must obey the protocol requirements for hosts when receiving packets from, and interacting with neighbors over, the latter (non-forwarding) interfaces.

3.2.3 IPv6 Header Format

Figure 3.2.3 depicts the IPv6 Header format. The fields in the header have the following meanings:

- **Version**: 4-bit Internet Protocol version number = 6.
- **Traffic Class**: 8-bit traffic class field.
- **Flow Label**: 20-bit flow label.
- **Payload Length**: 16-bit unsigned integer. Length of the IPv6 payload, that is, the rest of the packet following this IPv6 header, in octets. (Note that any extension headers present are considered part of the payload, that is, included in the length count)
- **Next Header**: 8-bit selector. Identifies the type of header immediately following the IPv6 header. Uses the same values as the IPv4 Protocol field.
- **Hop Limit**: 8-bit unsigned integer. Decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.
- **Source Address**: 128-bit address of the originator of the packet. This is covered later in more detail.
- **Destination Address**: 128-bit address of the intended recipient of the packet (possibly not the ultimate recipient, if a Routing header is present).

3.2.4 IPv6 Extension Headers

In IPv6, optional Internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper-layer header in a packet. There are a small number of such extension headers, each identified by a distinct Next Header value. As illustrated in the examples of Figure 3.2.4, an IPv6 packet may carry zero, one, or more extension headers, each identified by the Next Header field of the preceding header.

With one exception, extension headers are not examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in case of multicast) identified in the Destination address field of the IPv6 header. There, normal demultiplexing on the Next Header field of the IPv6 header invokes the module to process the first extension header, or the upper layer header if no extension header is present. The contents and semantics of each extension header
determine whether or not to proceed to the next header. Therefore, extension headers must be processed strictly in the order they appear in the packet. A receiver must not, for example, scan through a packet looking for a particular kind of extension header and process that header prior to processing all preceding ones.

The exception referred to in the preceding paragraph is the Hop-by-Hop Options header, which carries information that must be examined and processed by every node along a packet's delivery path, including the source and destination nodes. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header; its presence is indicated by the value zero in the Next Header field of the IPv6 header.

If, as a result of processing a header, a node is required to proceed to the next header but the Next Header value in the current header is unrecognized by the node, it should discard the packet and send an ICMP Parameter Problem message to the source of the packet, with an ICMP Code value of 1 ("unrecognized Next Header type encountered") and the ICMP Pointer field containing the offset of the unrecognized value within the original packet. The same action should be taken if a node encounters a Next Header value of zero in any header other than an IPv6 header.

Each extension header is an integer multiple of 8 octets long, in order to retain 8-octet alignment for subsequent headers. Multi-octet fields within each extension header are aligned on their natural boundaries, that is, fields of width n octets are placed at an integer multiple of n octets from the start of the header, for n = 1, 2, 4, or 8.

A full implementation of IPv6 includes implementation of the following extension headers:

- Hop-by-Hop Options
- Routing (Type 0)
- Fragment
- Destination Options
- Authentication
- Encapsulating Security Payload

The first four are specified in this RFC; the last two are specified in RFC 2402 and RFC 2406, respectively.

```
+-----------------------------------------------+-----------------------------------------------+-----------------------------------------------+-----------------------------------------------+-----------------------------------------------+
| IPv6 Header | Routing Header | Fragment Header | Fragment of TCP |
| Next Header = Routing | Next Header=Fragment | Next Header = TCP | Header + Data |
```

Fig. 3.2.4 Example of extension headers.

### 3.2.5 IPv6 Infrastructure

The IPv6 Specification (RFC 2460) and the IPv6 Addressing Architecture (RFC 2373) provide the base architecture and design of IPv6; we covered some of these key concepts in earlier sections. Here we look at basic IPv6 network constructs, specifically routing processes. Because there are differences on some of the details of how these IPv6 processes operate compared with IPv4, it is worth looking at some of these issues. Related work in IPv6 that needs to be mastered by implementers and network designers (covered in chapters that follow) includes the IPv6 Stateless Address Autoconfiguration (RFC 2462); the IPv6 Neighbor Discovery (NO) Processing (RFC 2461); the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) (RFC 3315); and, the Dynamic Updates to DNS (RFC 2136). Some portions of this discussion are based on [10].
Protocol Mechanisms
As we discussed earlier, the IPv6 header consists of two parts: the IPv6 base header, and optional extension headers. The optional extension headers are considered part of the IPv6 payload, as are the TCP/UDP/RTP PDUs. Obviously, IPv4 headers and IPv6 headers are not automatically interoperable; hence, a router operating in a mixed environment must support an implementation of both IPv4 and IPv6 in order to deal with both header formats. Figure 3.2.5 shows for illustration purposes the flows of IPv6 PDUs in a VoIP environment.

As we noted in passing in the previous chapter, the large size of the IPv6 address allows it to be subdivided into hierarchical routing domains that are supportive of the topology of today’s ubiquitous Internet (IPv4-based Internet lacks this flexibility). Conveniently, the use of 128 bits provides multiple levels of hierarchy and flexibility in designing hierarchical addressing and routing.

Protocol-Support Mechanisms
Two support mechanisms are of interest: (i) a mechanism to deal with communication transmission issues; and (ii) a mechanism to support multicast Internet Control Message Protocol for IPv6 (ICMPv6) (defined in RFC 2463) is designed to enable hosts and routers that use IPv6 protocols to report errors and forward along other basic status messages. For example, ICMPv6 messages are sent by Network Elements when an IPv6 POU cannot be forwarded further along to reach its intended destination. ICMPv6 messages are carried as the payload of IPv6 POUs; hence, there is no guarantee on their delivery.

The following list identifies the functionality supported by the basic ICMPv6 mechanisms:
3.2.6 ICMPv6 message

The ping command is basically an ICMPv6 Echo Request message along with the receipt of an ICMPv6 Echo Reply message. Just as is the case with IPv4, one can use pings to detect network or host communication failures and troubleshoot connectivity problems.

ICMPv6 also supports Multicast Listener Discovery (MLD) mechanism. MLD (RFC 2710, RFC 3590, and RFC 3810) enables one to manage subnet multicast membership for IPv6. MLD is a collection of three ICMPv6 messages that replace the Internet Group Management Protocol (IGMP) version 3 that is employed in IPv4. MLD messages are used to determine group membership on a network segment, also known as a link or subnet. As implied, MLD messages are sent as ICMPv6 messages. They are used in the context of multicast communications (see below):

- Multicast Listener Query: Message issued by a multicast router to poll a network segment for group members. Queries can be general, requesting group membership for all groups, or can request group membership for a specific group.
- Multicast Listener Report: Message issued by a host when it joins a multicast group, or in response to an MLD Multicast Listener Query sent by a router.
- Multicast Listener Done: Message issued by a host when it leaves a host group and is the last member of that group on the network segment.

ICMPv6 also supports Neighbor Discovery (ND). ND (RFC 2461) is a collection of five ICMPv6 messages that manage node-to-node communication on a link. Nodes on the same link are also called neighboring nodes. ND replaces Address Resolution Protocol (ARP), ICMPv4 Router Discovery, and the ICMPv4 Redirect message. Hosts, servers, SIP proxies, H.323 gatekeepers, etc. use ND to discover neighboring routers, addresses, address prefixes, and other configuration parameters. Routers make use of ND to advertise their presence, host configuration parameters, and on-link prefixes. Routers also use ND to inform hosts of a better next-hop address to forward PDUs for a specific destination. Nodes make use of ND to resolve the link-layer address of a neighboring node to which an IPv6 PDU is being forwarded. Nodes also use ND to determine when the link-layer address of a neighboring node has changed and whether IPv6 PDUs can be sent to and received from a neighbor.

3.2.7 Dynamic Host Configuration Protocol for IPv6

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6 or more simply DHCP) enables DHCP servers to pass configuration parameters such as IPv6 network addresses to IPv6 nodes. DHCP provides both robust stateful auto-configuration and auto-registration of DNS Host Names:

DHCPv6 also defines the IPv4-compatible IPv6 address, e.g.,::156.55.23.5, discussed previously in the section on IPv6 address space. IPv4-compatible IPv6 addresses are used to implement a simple automatic tunneling mechanism.

In addition to connectivity issues at the IP layer, the transition to IPv6 is also not entirely transparent to the networking layers above IP. As discussed previously, IPv6 addresses are significantly longer in size than IPv4 addresses and thus will require a change in application programming interfaces (APIs) or service primitive parameters that include IP addresses. Applications must also be extended to select the appropriate
protocol, IPv4 or IPv6, when a DNS lookup returns both types of addresses. In general, legacy applications written for IPv4 either need to be rewritten or amended to support IPv6. For example, the application layer file transfer protocol (FTP) embeds IP addresses in its protocol fields, and could thus require changes to both the client and server FTP applications.

The IETF has defined a number of specific mechanisms to assist in transitioning to IPv6. These mechanisms are generally classified as belonging to the following categories:

Note that a transition mechanism may employ techniques from more than one of these categories. For example, when an end system or router creates an IPv6 in IPv4 tunnel, this could be classified as both dual stack (having both an IPv4 and IPv6 address) and tunneling.

### 3.2.8 Neighbor Discovery for IP Version 6 (IPv6) Protocol

This section provides a brief overview of the Neighbor Discovery for IP Version 6 (IPv6) protocol, also known as Neighbor Discovery Protocol (NDP) based directly on RFC 2461.

Neighbor Discovery for IP Version 6 (IPv6) protocol solves a set of problems related to the interaction between nodes attached to the same link. It defines mechanisms for solving each of the following problems:

### 3.2.9 Mobile IP Version 6 (MIPv6)

This section provides a short introduction to MIPv6, based directly on RFC 3775. RFC 3775 is a lengthy RFC due to the complexity of the topic. Only the most basic capabilities are covered here, and the potential investigator and researcher is encouraged to refer to the RFC for complete coverage.

MIPv6 is a protocol that allows nodes to remain reachable while moving around in the IPv6 Internet. Each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a care-of address, which provides information about the mobile node's current location. IPv6 packets addressed to a mobile node's home address are transparently routed to its care-of address. The protocol enables IPv6 nodes to cache the binding of a mobile node's home address with its care-of address, and to then send any packets destined for the mobile node directly to it at this care-of address. To support this operation, MIPv6 defines a new IPv6 protocol and a new destination option. All IPv6 nodes, whether mobile or stationary, can communicate with mobile nodes. RFC 3775 specifies a protocol, known as Mobile IPv6, that allows nodes to remain reachable while moving around in the IPv6 Internet. Without specific support for mobility in IPv6, packets destined to a mobile node would not be able to reach it while the mobile node is away from its home link. In order to continue communication in spite of its movement, a mobile node could change its IP address each time it moves to a new link, but the mobile node would then not be able to maintain transport and higher-layer connections when it changes location. Mobility support in IPv6 is particularly important, as mobile computers are likely to account for a majority or at least a substantial fraction of the population of the Internet during the lifetime of IPv6. The protocol defined in the RFC allows a mobile node to move from one link to another without changing the mobile node's "home address." Packets may be routed to the
mobile node using this address regardless of the mobile node's current point of attachment to the Internet. The mobile node may also continue to communicate with other nodes (stationary or mobile) after moving to a new link. The movement of a mobile node away from its home link is thus transparent to transport and higher-layer protocols and applications.

3.3 Performance Issues in High Speed (Gigabit) Networks

3.3.1 Introduction
One of the fundamental characteristics which distinguishes gigabit(Gb) networks from traditional Mb lines is that long-distance Gb lines are delay-limited rather than bandwidth-limited.

3.3.2. Performance & Reliability Issues
Early attempts at using Gb networks involved the use of older, existing protocols without any modifications. However, this approach quickly resulted in several performance bottlenecks and issues related to overall quality of service and capacity utilization. The following sections discuss some of these problems in detail.

(a) Gigabit lines are delay-limited
The most fundamental difference between traditional megabit lines and Gb lines is that the latter are delay-limited rather than bandwidth-limited. Fig.3.3.2 depicts the time it takes to transfer a 1-Mb file 4000 km at various transmission speeds.

(b) Excessive network speed relative to computing speed
A second problem is that communication speeds have improved much faster than computing speeds. This is evident from the fact that at the time of the advent of the ARPANET, when packet size was about 1000 bits, the existing computers ran at 1mips, the ARPANET was capable of a packet delivery rate of 56 packets/sec.

(c) Bandwidth-Delay product negatively impacts older protocols
It is increasingly being observed that older, reliable-delivery protocols based on the Go-Back-N principle yield very low throughput rates on transmission lines with a large bandwidth-delay product.

(d) Use of 32-bit sequence numbers reduces reliability of packet sequencing
One of the pivotal but unstated assumptions in the design of packet-sequencing techniques used with older and current protocols is that the time to use up the entire sequence space would greatly exceed the maximum packet lifetime. As a result, there used to be no potential possibility of old duplicates still existing when the sequence numbers wrapped around and reset.
(e) Multimedia application require predictable and consistent data rate
For many new applications such as multimedia, which harness the capability of Gigabit networks, two equally important factors impact their performance.

3.3.3 Potential Solutions & Alternatives
(a) Efficient Protocol Mechanisms
Older and existing protocols have been designed with the objective of minimizing the number of bits on the wire, frequently using small fields and packing them together into bytes and words. H

(b) Packet Layout Schemes
The protocols for Gb networks must be designed with special consideration toward the layout and size of the fields as well as the degree of error checking.

(c) Protocol Software Improvements
Interrupt processing [4] can be a major bottleneck in the end-to-end performance of Gigabit networks. The performance of Gigabit network end hosts or servers can be severely degraded due to interrupt overhead caused by heavy incoming traffic. In particular, excessive latency and significant degradation in system throughput can be encountered. Also, user applications may livelock as the CPU power gets mostly consumed by interrupt handling and protocol processing. A number of interrupt-handling schemes has been proposed and employed to mitigate the interrupt overhead and improve OS performance. Among the most popular interrupt-handling schemes are normal interruption, polling, interrupt coalescing, and disabling and enabling of interrupts. In previous work, we presented a preliminary analytical study and models of normal interruption and interrupt coalescing. In this article, we extend our analysis and modeling to include polling and the scheme of interrupt disabling and enabling. For polling, we study both pure (or FreeBSD-style) polling and Linux NAPI polling.

3.3.4. Conclusions
Network performance is typically dominated by protocol processing overhead and this situation gets worse at higher speeds. Protocols should be designed to minimize the number of TPDUs, context switches and the number of times each TPDU is copied. For Gb networks, simple protocols are desirable.

References


RFCs