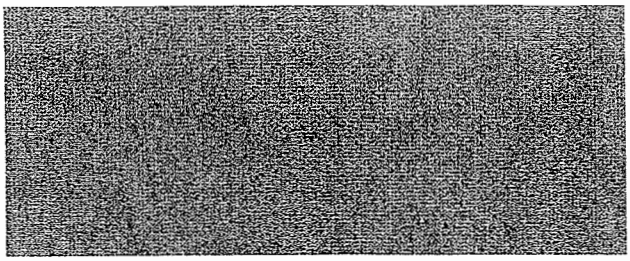
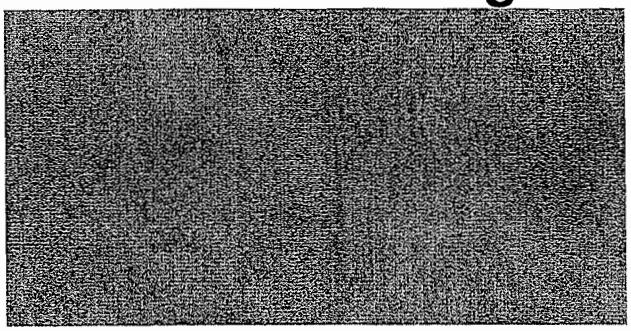
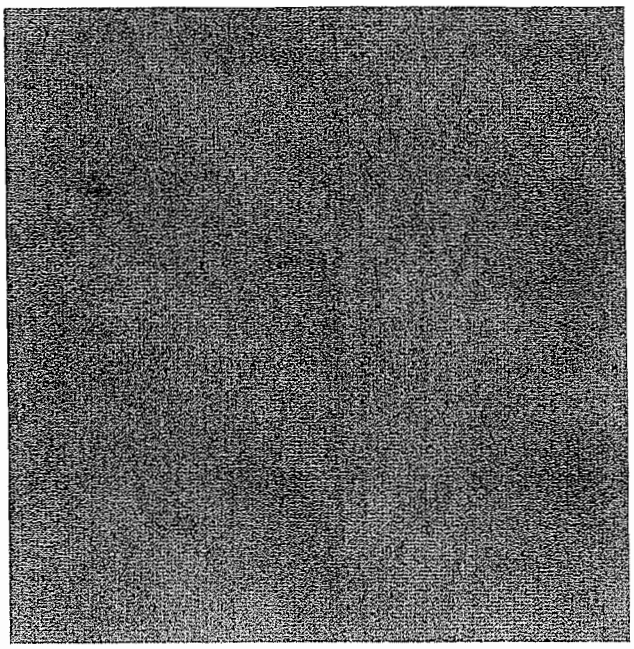


IP Networking



STUDENT BOOK
LZT 123 7773 R2A



DISCLAIMER

This book is a training document and contains simplifications. Therefore, it must not be considered as a specification of the system.

The contents of this document are subject to revision without notice due to ongoing progress in methodology, design and manufacturing.

Ericsson assumes no legal responsibility for any error or damage resulting from the usage of this document.

This document is not intended to replace the technical documentation that was shipped with your system. Always refer to that technical documentation during operation and maintenance.

This document was produced by Ericsson AB.

- It is used for training purposes only and may not be copied or reproduced in any manner without the express written consent of Ericsson.

This Student Book, LZT 123 7773, R2A supports course number LZU 102 397 .

Table of Contents

Intentionally Blank

1 IP Networking

After completing this chapter you will be able to:

- Describe the history of TCP/IP
- Outline the history of the Internet
- Outline the different functions of the Internet organisations (IAB, ICANN, IETF, IRTF, IANA)
- Describe request for comments (RFCs)
- Outline the OSI 7-layer model
- Outline the TCP/IP protocol stack

Intentionally Blank

HISTORY OF TCP/IP

TCP/IP was initially designed to meet the data communication needs of the U.S. Department of Defence (DOD). In the late 1960s the Advanced Research Projects Agency (ARPA, now called DARPA) of the U.S. Department of Defense began a partnership with U.S. universities and the corporate research community to design open, standard protocols and build multi-vendor networks.

Together, the participants planned ARPANET, the first packet switching network. The first experimental four-node version of ARPANET went into operation in 1969. These four nodes at three different sites were connected together via 56 kbit/s circuits, using the Network Control Protocol (NCP). The experiment was a success, and the trial network ultimately evolved into a useful operational network, the "ARPA Internet".

In 1974, Vinton G. Cerf and Robert E. Kahn proposed the design for a new set of core protocols, for the ARPANET, in a paper. The official name for the set of protocols was TCP/IP Internet Protocol Suite, commonly referred to as TCP/IP, which is taken from the names of the network layer protocol (Internet protocol [IP]) and one of the transport layer protocols (Transmission Control Protocol [TCP]).

TCP/IP is a set of network standards that specify the details of how computers communicate, as well as a set of conventions for interconnecting networks and routing traffic. The initial specification went through four early versions, culminating in version 4 in 1979.

By 1979, so many researchers were involved in the TCP/IP effort that ARPA formed an informal committee to co-ordinate and guide the design of the protocols and architecture of the emerging Internet. Called the Internet Control and Configuration Board (ICCB), the group met regularly until 1983, when it was re-organised and renamed as the Internet Activities Board (IAB).

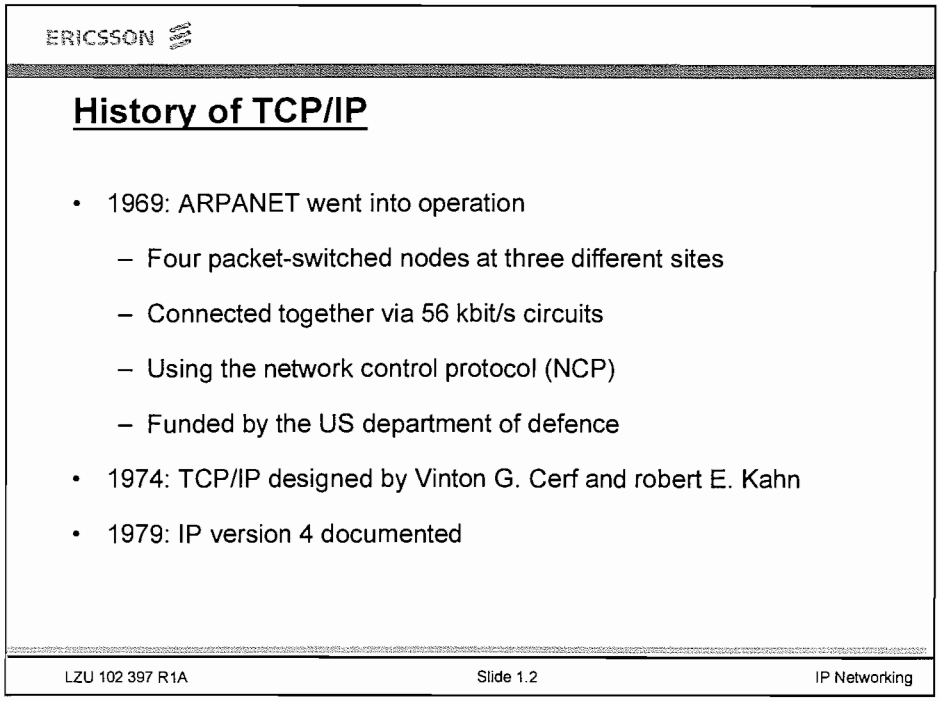


Figure 1-1.

Notes:

The global Internet began around 1980 when ARPA started converting machines attached to its research networks to the new TCP/IP protocol. The ARPANET already in place quickly became the backbone of the new Internet and was used for many of the early experiments with TCP/IP. The transition to Internet technology became complete in 1983, when the Defense Communications Agency (DCA) mandated that all computers connected to ARPANET use TCP/IP, which replaced the earlier Network Control Protocol (NCP). At the same time the DCA split the ARPANET into two separate networks, one for further research and one for military communication. The research part retained the name ARPANET; the military part, became known as MILNET.

To encourage university researchers to adopt and use the new protocols, ARPA made an implementation available at low cost. In 1979, most university computer departments were running a version of the UNIX operating system available in the University of California's Berkley Software Distribution (BSD).

By funding Bolt Beranek and Newman Inc. (BBN) to implement its TCP/IP protocols for use with UNIX, and funding Berkley to integrate the protocols with its software distribution, ARPA was able to supply the TCP/IP protocol to over 90% of the university computer science departments.

By 1985, the ARPANET was heavily used and congested. In response, the National Science Foundation (NSF) initiated phase one development of the NSFNET. ARPANET was officially decommissioned in 1989. The NSFNET was composed of multiple regional networks and peer networks (such as the NASA Science Network) connected to a major backbone that constituted the core of the overall NSFNET in its earliest form.

In 1991 the NSF decided to move the backbone to a private company and start charging institutions for connections. In 1991, Merit, IBM, and MCI started a not-for-profit company named Advanced Networks and services (ANS).

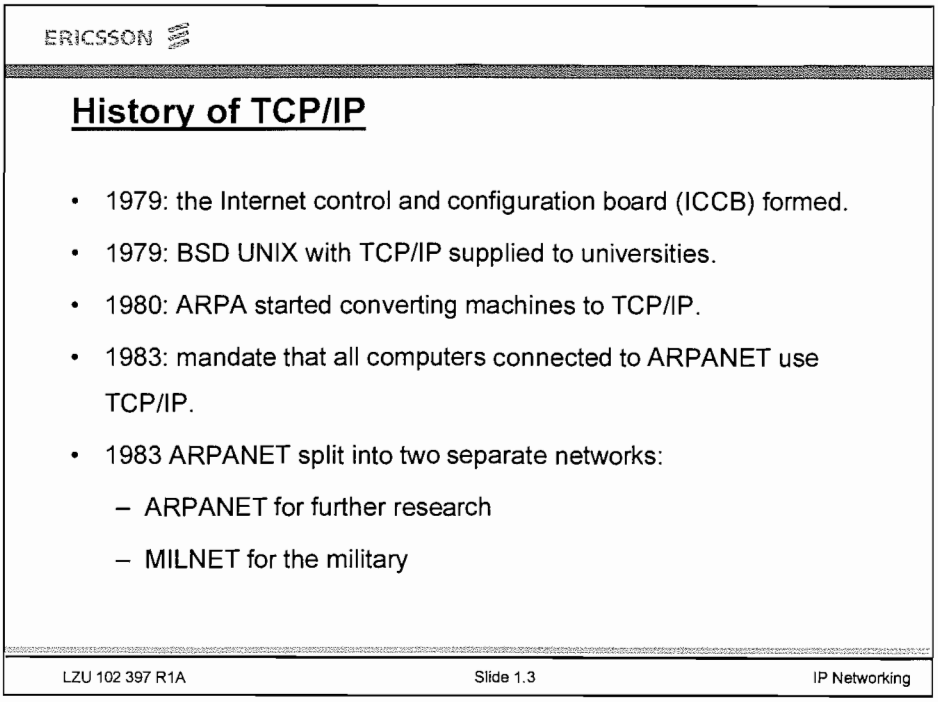


Figure 1-2.

Notes:

By 1993, ANS had installed a new network that replaced NFSNET. Called ANSNET, the new backbone operated over T3 (45 Mbit/s) links. ANS owned this new Wide Area Network (WAN), unlike previous WANs used in the Internet, which had all been owned by the U.S. government. In 1993, NSF invited proposals for projects to accommodate and promote the role of commercial service providers and lay down the structure of a new and robust Internet model. At the same time, NSF withdrew from the actual operation of the network and started to focus on research aspects and initiatives. The "NSF solicitation" included four separate projects for which proposals were invited:

Creating a set of Network Access Points (NAPs) where major providers connect their networks and exchange traffic.

Implementing a Route Arbiter (RA) project, to provide equitable treatment of the various network service providers with regard to routing administration.

Providing a very high-speed Backbone Network Service (vBNS) for educational and governmental purposes.

Moving existing "regional" networks, from the NSFNET backbone to other Network Service Providers (NSPs) which have connections to NAPs.

Partly as a result of the NSF solicitations, today's Internet structure has moved from a core network (NSFNET) to a more distributed architecture operated by commercial providers such as Sprint, MCI, BBN, and others connected via major network exchange points, called Network Access Points (NAPs).

A NAP is defined as a high-speed switch to which a number of routers can be connected for the purpose of traffic exchange. This allows Internet traffic from the customers of one provider to reach the customers of another provider.

Internet Service Providers (ISPs) are companies that provide Internet services, for example, Web access and Internet mail, to end customers, both individual users and corporate users. The connection point between a customer and an ISP is called a point of presence (POP).

The physical connection between a customer and as ISP can be provided by many different physical access methods, for example dial-up or Frame Relay. ISP networks exchange information with each other by connecting to NSPs that are connected to NAPs, or by connecting directly to NAPs.

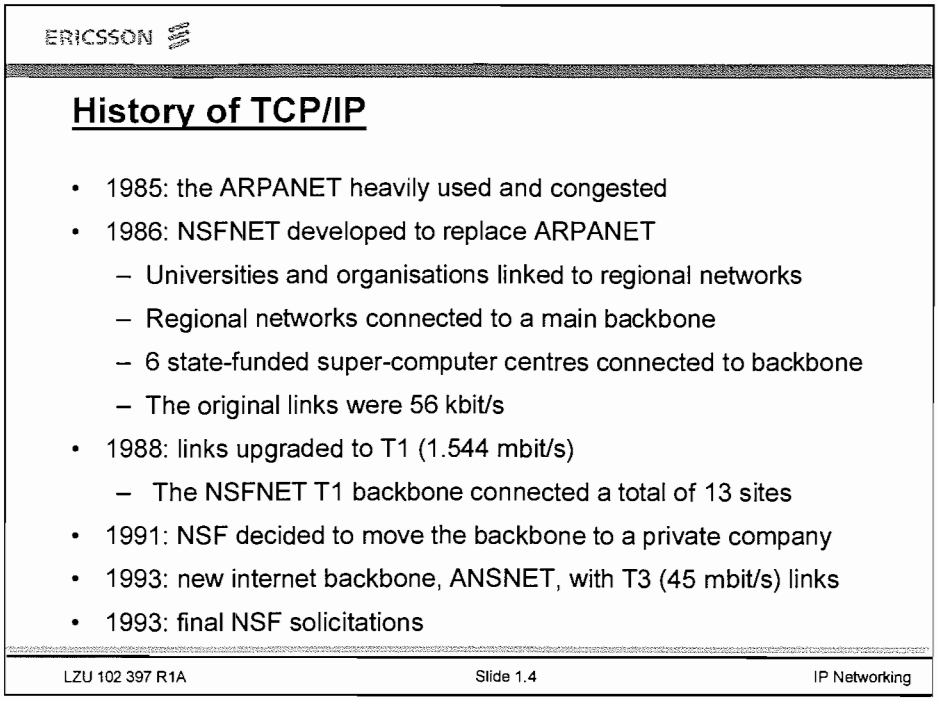


Figure 1-3.

Notes:

The NSFNET was physically connected to four NAPS between 1993 and 1995. Additional NAPs continue to be created around the world as providers keep finding the need to interconnect.

In 1995 the NSF awarded MCI the contract to build the very high performance Backbone Network Service (vBNS) to replace ANSNET. The vBNS was designed for the scientific and research communities with high-bandwidth uses and is not used for general Internet traffic. vBNS+ is a specialized nationwide IP network that supports high-performance, high-bandwidth applications. Originating as the very high performance Backbone Network Service (vBNS), vBNS+ is the product of a five-year cooperative agreement between MCI and the National Science Foundation.

See http://www.vbns.net for the latest information

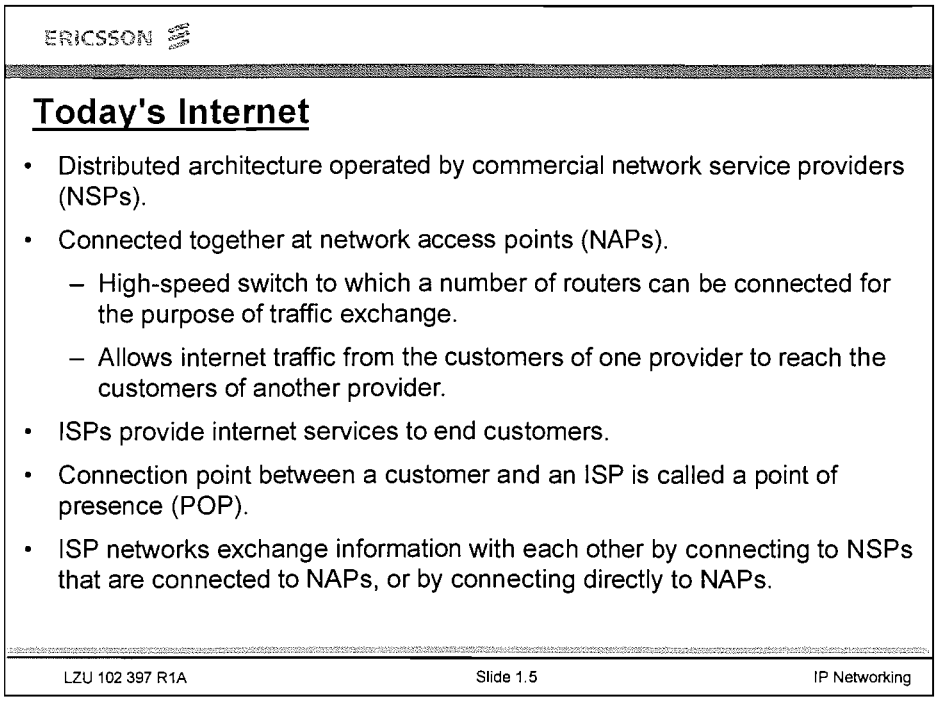


Figure 1-4.

Notes:

THE MAIN BODIES

The Internet SOCiety (ISOC)

The Internet Society is the international organization for global cooperation and coordination for the Internet and it's internetworking technologies and applications. The Internet Society is a professional membership organization of Internet experts that comments on policies and practices and oversees a number of other boards and task forces dealing with network policy issues.

Internet Architecture Board (IAB)

The IAB is responsible for defining the overall architecture of the Internet, providing guidance and broad direction to the IETF. The IAB also serves as the technology advisory group to the Internet Society, and oversees a number of critical activities in support of the Internet.

Internet Engineering Task Force (IETF)

The IETF is the protocol engineering and development arm of the Internet. Though it existed informally for some time, the group was formally established by the **Error! Hyperlink reference not valid.**IAB in 1986.

The Internet Engineering Steering Group (IESG)

The IESG is responsible for technical management of IETF activities and the Internet standards process. As part of the ISOC, it administers the process according to the rules and procedures, which have been ratified by the ISOC Trustees. The IESG is directly responsible for the actions associated with entry into and movement along the Internet "standards track," including final approval of specifications as Internet Standards.

Internet Assigned Numbers Authority (IANA)

Based at ICANN, IANA is in charge of all "unique parameters" on the Internet, including IP (Internet Protocol) addresses. Each domain name is associated with a unique IP address, a numerical name consisting of four blocks of up to three digits each, e.g. 204.146.46.8, which systems use to direct information through the network.

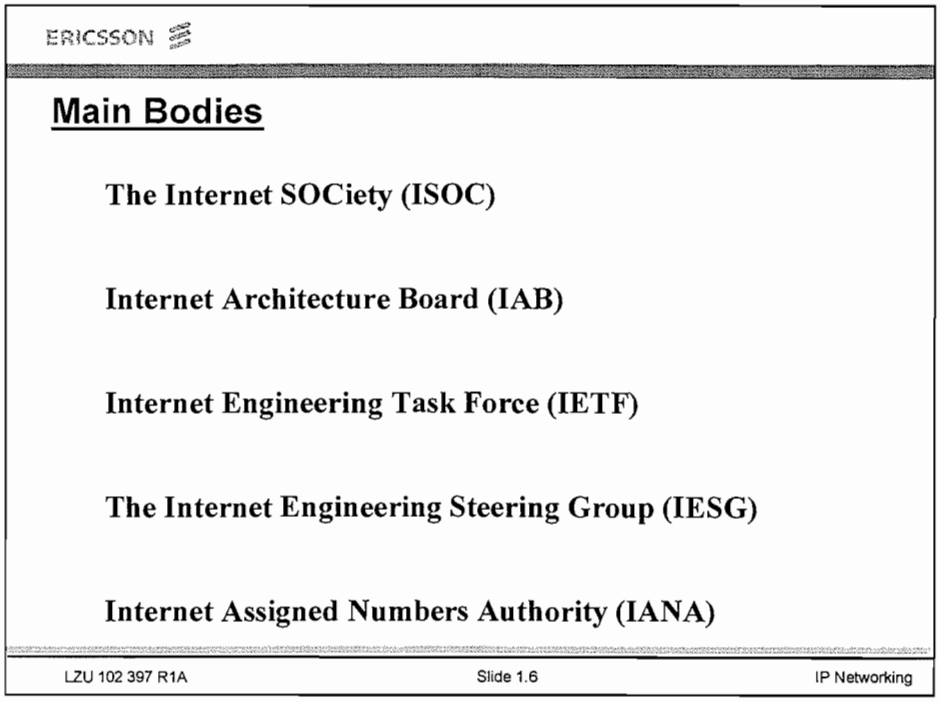


Figure 1-5.

Notes:

ISOC (INTERNET SOCIETY)

The Internet Society is an international, non-profit, membership organization that fosters the expansion of the Internet. One of the ways that ISOC does this is through financial and legal support of the other "I" groups described here, particularly the IETF. ISOC's oversight of the IETF is remarkably hands-off, so many IETF participants don't even know about it.

ISOC provides insurance coverage for many of the people in the IETF process, and acts as a public relations channel for the times that one of the "I" groups wants to say something to the press.

The ISOC is one of the major unsung (and under-funded) heroes of the Internet. Its members reflect the breadth of the entire Internet community and consist of individuals, corporations, non-profit organizations, and government agencies.

Its principal purpose is to maintain and extend the development and availability of the Internet and its associated technologies and applications - both as an end in itself, and as a means of enabling organizations, professions, and individuals worldwide to more effectively collaborate, cooperate, and innovate in their respective fields and interests.

The Internet Society was announced in June 1991 at an international networking conference in Copenhagen and brought into existence in January 1992 by a worldwide cross-section of individuals and organizations who recognized that the Society was a critical component necessary to evolve and globalize the Internet and internet technologies and applications, and to enhance their availability and use on the widest possible scale.

Its specific goals and purposes include:

- Development, maintenance, evolution, and dissemination of standards for the Internet and its internetworking technologies and applications;
- Growth and evolution of the Internet architecture;
- Maintenance and evolution of effective administrative processes necessary for operation of the global Internet and internets;
- Education and research related to the Internet and internetworking;
- Harmonization of actions and activities at international levels to facilitate the development and availability of the Internet;
- Collection and dissemination of information related to the Internet and internetworking, including histories and archives;
- Assisting technologically developing countries, areas, and peoples in implementing and evolving their Internet infrastructure and use;
- Liaison with other organizations, governments, and the general public for coordination, collaboration, and education in effecting the above purposes.

INTERNET ARCHITECTURE BOARD (IAB)

The IAB is the coordinating committee for Internet design, engineering and management. The IAB has a maximum of 15 members who work on a voluntary basis. The IAB was reorganised in 1992 when it was brought under the auspices of the Internet Society (ISOC). The IAB was renamed the Internet Architecture Board, but the functions remained reasonably unchanged. Individuals are nominated for membership to the IAB by Internet community members and selected by the ISOC trustees for two-year, renewable terms.

The IAB creates task forces, committees, and working groups as required within the scope of the IAB's responsibility. The initial appointments are the following: the editor of the RFC publication series and the chairs of the IETF and the IRTF.

Members of the IAB appoint the chair of the IAB who then has the authority to organise and direct task forces as deemed necessary. The Internet Engineering Steering Group (IESG) members are nominated by the Internet community and selected by the IAB. All terms are two years renewable.

The chairman and the IESG members organise and manage the IETF. There is an overlap of functions and membership between the IETF and the IRTF, with the major difference being viewpoint and sometimes time frame. This overlap is deliberate and considered vital for technology transfer.

The individual members of the ISOC elect trustees for three-year terms. Volunteers manage the infrastructure of the ISOC, including members of the IAB and its task forces. Although several government agencies continue to support key aspects of the TCP/IP protocol development, the majority of personal activity (for example, attending meetings writing Request For Comments RFCs) is done on a voluntary basis.

IAB responsibilities include:

- IESG Confirmation: The IAB confirms the IETF Chair and IESG Area Directors, from nominations provided by the IETF Nominating Committee.
- Architectural Oversight: The IAB provides oversight of, and occasional commentary on, aspects of the architecture for the protocols and procedures used by the Internet.
- Standards Process Oversight and Appeal: The IAB provides oversight of the process used to create Internet Standards. The IAB serves as an appeal board for complaints of improper execution of the standards process through acting as an appeal body in respect of an IESG standards decision.
- RFC Series and IANA: The IAB is responsible for editorial management and publication of the Request for Comments (RFC) document series, and for administration of the assignment of IETF Protocol parameter values by the IETF Internet Assigned Numbers Authority (IANA).
- External Liaison: The IAB acts as representative of the interests of the IETF in liaison relationships with other organizations concerned with standards and other technical and organizational issues relevant to the world-wide Internet.
- Advice to ISOC: The IAB acts as a source of advice and guidance to the Board of Trustees and Officers of the Internet Society concerning technical, architectural, procedural, and (where appropriate) policy matters pertaining to the Internet and its enabling technologies.
- IRTF Chair: The IAB selects a chair of the Internet Research Task Force (IRTF) for a renewable two year term.

INTERNET ENGINEERING TASK FORCE (IETF)

The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual. The IETF does not deal with the operation of any Internet network, nor does it set any operational policies. Its charter is to specify the protocols and architecture of the Internet and recommend standards for IAB approval.

The actual technical work of the IETF is done in its working groups, which are organized by topic into several areas (e.g., routing, transport, security, etc.). The IETF holds meetings three times per year.

The IETF working groups are grouped into areas, and managed by Area Directors, who are members of the Internet Engineering Steering Group (IESG). The IAB provides architectural oversight. The IAB also adjudicates appeals when someone complains that the IESG has failed. The IAB and IESG are chartered by the Internet Society (ISOC) for these purposes. The General Area Director also serves as the chair of the IESG and of the IETF, and is an ex-officio member of the IAB.

The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols. The IANA is chartered by the Internet Society (ISOC) to act as the clearinghouse to assign and coordinate the use of numerous Internet protocol parameters.

Its mission includes:

- Identifying, and proposing solutions to pressing operational and technical problems in the Internet.
- Specifying the development or usage of protocols and the near-term architecture to solve technical problems for the Internet
- Making recommendations to the Internet Engineering Steering Group (IESG) regarding the standardization of protocols and protocol usage in the Internet.
- Facilitating technology transfer from the Internet Research Task Force (IRTF) to the wider Internet community.
- Providing a forum for the exchange of information within the Internet community between vendors, users, researchers, agency contractors, and network managers.

INTERNET RESEARCH TASK FORCE (IRTF)

The Research Groups work on topics related to Internet protocols, applications, architecture and technology. Research Groups are expected to have the stable long-term (with respect to the lifetime of the Research Group) membership needed to promote the development of research collaboration and teamwork in exploring research issues. Participation is by individual contributors, rather than by representatives of organizations.

The IRTF is managed by the IRTF Chair in consultation with the Internet Research Steering Group (IRSG). The IRSG membership includes the IRTF Chair, the chairs of the various Research Group and possibly other individuals ("members at large") from the research community.

The IRTF Chair is appointed by the Internet Architecture Board (IAB). The Research Group chairs are appointed as part of the formation of Research Groups and the IRSG members at large are chosen by the IRTF Chair in consultation with the rest of the IRSG and on approval of the IAB. In addition to managing the Research Groups, the IRSG may from time to time hold topical workshops focusing on research areas of importance to the evolution of the Internet, or more general workshops to, for example, discuss research priorities from an Internet perspective.

The IRTF Research Groups guidelines and procedures are described more fully in RFC 2014 (BCP 8).

The IRTF is concerned with understanding technologies and how they may be used in the Internet, rather than products or standard protocols. However, specific experimental protocols may be developed, implemented and tested to gain the required understanding.

INTERNET ASSIGNED NUMBER AUTHORITY (IANA)

The Internet employs a central Internet Assigned Numbers Authority (IANA) for the allocation and assignment of various numeric identifiers needed for the operation of the Internet. The IANA function is performed by the University of Southern California's Information Sciences Institute. The IANA is chartered by the IAB to co-ordinate the assigned values of protocol parameters, including type codes, protocol numbers, port numbers, Internet addresses, and Ethernet addresses.

The IANA delegates the responsibility of assigning IP network numbers and domain names to four Regional Internet Registries (RIRs):

- Asia Pacific Network Information Centre (http://www.apnic.net) (APNIC)
- The American Registry for Internet Numbers (http://www.arin.net) (ARIN)
- Latin American and Caribbean Internet Addresses Registry (http://www.lacnic.net) (LACNIC)
- Réseaux IP Européens Network Coordination Centre (http://www.ripe.net) (RIPE NCC)

The registries provide databases and information servers such as WHOIS registry for domains, networks, AS numbers, and their associated Point Of Contacts (POCs). The documents distributed by the Internet registries include network information, and procedures, including application forms, to request network numbers and register domain name servers. All of these are available from the relevant web sites.

The Regional Internet Registries communicate to the Government Advisory Committee (GAC) of the Internet Corporation for Assigned Names and Numbers (ICANN) their view of the roles and responsibilities of the GAC with regard to Internet Number Resources.

REGIONAL INTERNET REGISTRIES

The Regional Internet Registries (RIRs) are responsible for a critical component in the operational infrastructure of the Internet. They execute this responsibility by jointly undertaking the role of management of Internet Number Resources through the allocation of Internet Protocol addresses (currently IPv4 and IPv6) and the identifiers used in Internet inter-domain routing (currently Border Gateway Protocol Autonomous System Numbers (ASNs)) to network operators and Local Internet Registries.

The RIRs manage the part of the DNS name space that pertains to these Internet Number Resources (currently contained within in-addr.arpa and ip6.arpa). These managerial roles are in support of the ultimate requirement within the Internet to associate network resources with numbers drawn from the relevant public Internet number space.

The objectives of Internet Number Resource management are those of responsible stewardship of the resource, ensuring that the resources are managed fairly, uniformly, and that there is long-term availability in all geographic areas for present and future operators and users of the Internet. The RIRs perform this today, using a regional delineation of areas of responsibility on an approximate continental scale.

The establishment of Regional Internet Registries was initiative by the IETF in 1992 (through RFC1466, which proposes implementation of the recommendations of RFC1174 (1990)). RIPE NCC and APNIC were established soon afterwards, in 1992 and 1993 respectively, and have undertaken their function ever since. ARIN exists today as a continuation of the original InterNIC, while LACNIC was established in 2002. The establishment of three of the RIRs therefore precedes that of ICANN and the GAC by some 6 years.

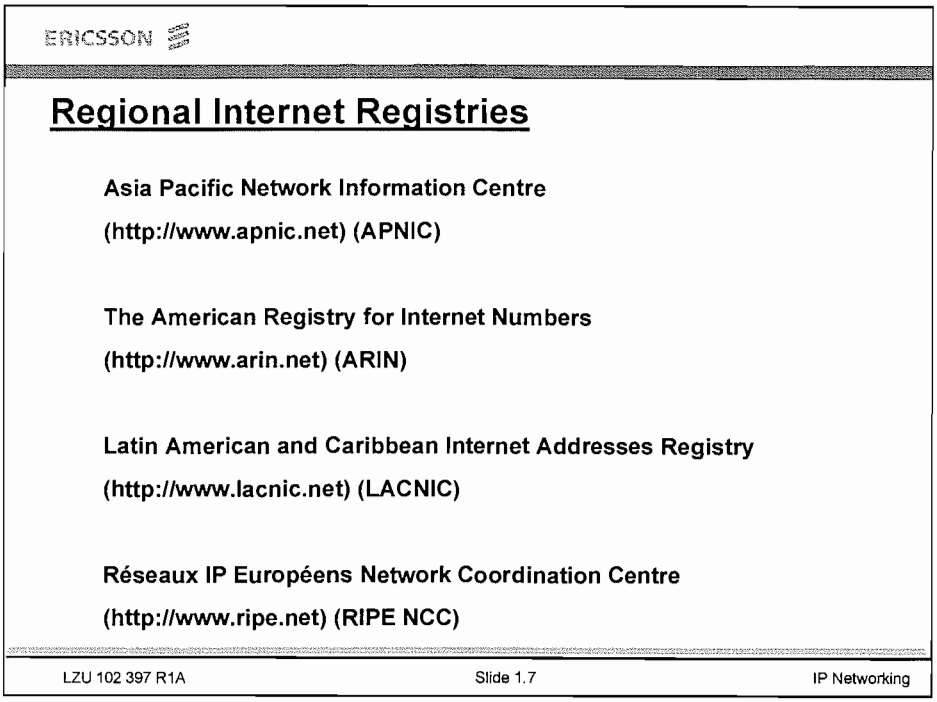


Figure 1-6.

Notes:

The four existing RIRs are set up as not-for-profit organizations whose membership is open to all interested parties. The membership of the RIRs consists of thousands of Internet operators (ISPs) and other companies and stakeholders. Each RIR has an Executive Board that is elected by the membership. The activities of the RIR are approved by the membership.

The RIRs do not charge for Internet Number Resources. There is an initial and annual fee for the services that are required to support the management of the Internet Number Resources. These services include the following, which are conducted within a structured industry self-regulatory framework:

- Allocation of Internet Protocol addresses (currently IPv4 and IPv6);
- Allocation of identifiers used in Internet inter-domain routing (currently Border Gateway Protocol autonomous system numbers (ASNs));
- Provision and maintenance of the part of the DNS name space that pertains to these Internet Number Resources (currently contained within in-addr.arpa and ip6.arpa;
- Provision and maintenance of WHOIS information directory;
- Provision and maintenance of Internet Routing Registry information;

In addition to these services, the RIRs contribute to the ICANN budget and completely fund all activities of the ICANN Address Supporting Organization (ASO). All RIR corporate reports, including financial information, are publicly available on each RIR web site.

APNIC is the only RIR with an active National Internet Registry (NIR) structure. After a period of some years during which new NIRs were no longer accepted by APNIC (due to policy and operational inconsistencies), the NIR membership structure was reopened in 2002. New NIRs will now be recognized by application to APNIC's Executive Board, providing that they have official Government sanction, and can demonstrate the ability and capacity to carry out delegated responsibilities in accordance with APNIC policy. It should be noted that NIRs do not have the exclusive right to perform resource distribution functions in their country or economy; any ISP may choose freely to receive resource services from APNIC or from the available NIR (where it exists).

ICANN.

The Internet Corporation for Assigned Names and Numbers (ICANN) is the non-profit corporation that was formed to assume responsibility for the IP address space allocation, protocol parameter assignment, domain name system management, and root server system management functions previously performed under U.S. Government contract by IANA and other entities.

ICANN is a private sector initiative to assume responsibility for overseeing the technical coordination of the Domain Name System (DNS), which allows Internet addresses (for example, web pages and email accounts) to be found by easy-to-remember names, instead of numbers.

Incorporated and headquartered in California, ICANN is a non-profit corporation structured to make decisions on the basis of Internet community consensus. As ICANN's start-up phase progresses, its Board of Directors will be elected in part by a global membership of individual members of the Internet community, and in part by supporting organizations representing the business, technical, non-commercial and academic communities.

ICANN represents an unprecedented effort by the Internet business, technical, non-commercial and academic communities to create a globally representative private sector (that is, non-governmental) policymaking body. Consistent with the principle of maximum self-regulation in the high-tech economy, ICANN is perhaps the foremost example of collaboration by the various constituents of the Internet community – individuals and organizations, entrepreneurs and educators, corporate enterprises and non-profit advocacy groups. Though often contentious, the ICANN structure creates an open and transparent global forum in which competing interests can work toward consensus.

The many hundreds of individuals, organizations, corporations, engineers, entrepreneurs, educators and others that have participated in the process of creating and building ICANN encompass a wide cross-section of the global Internet community. ICANN recognizes that it will require significant ongoing effort in the coming months to reach out and to bring previously uninvolved individuals and organizations into the ICANN process, particularly in the developing world.

To accomplish this, ICANN and the Department of Commerce (DoC) entered into a Memorandum of Understanding (MoU) on November 25, 1998, agreeing to work together to manage the transition from government control to private sector control. The single most visible and important element of DNS management is the registration of domain names (for example, .com, .org and .net). A single historical provider, Network Solutions, has for years enjoyed a government-granted monopoly over new domain name registrations and renewals.

Under the MoU, ICANN has already accredited a number of new competitive as part of a test of the Shared Registration System, which will permit competition among multiple registrars in this very public component of the Internet's underlying technology.

The move to ICANN-administered open competition in the market for domain name registrations, with the addition of dozens of accredited competitive registrars, is likely to see consumers reap significant price benefits far in excess of ICANN's administrative costs.

Building a global, consensus-driven organization has been a complicated and contentious task. The development of consensus from vastly diverse views and interests is a daunting challenge.

ICANN has no statutory or other governmental power: its authority is entirely a consequence of voluntary contracts and compliance with its consensus policies by the global Internet community. It has no power to force any individual or entity to do anything; its "authority" is nothing more than the reflection of the willingness of the members of the Internet community to use ICANN as a consensus development vehicle.

ICANN was created and has developed under the full scrutiny of the public eye. The agendas, results, and minutes of the Initial Board's deliberations are widely publicized, and posted in advance. The Board holds a quarterly public meeting where everything on the agenda is subject to full and open public discussion. In order to reduce costs for participants, ICANN broadcasts its public meetings live over the Internet, allowing remote participants to watch and send comments and questions by email to the meeting room. The text of all resolutions adopted by the Board is immediately released, and the Board holds a public press conference. All decisions of substance are preceded by prior notice and a full opportunity for public comment.

ICANN was created by members of the Internet community in response to a June 1998 White Paper, issued by the U.S. Department of Commerce (DOC). As the Internet developed, DNS functions were carried out by a variety of volunteers and US Government contractors. A non-competitive, government-funded system developed. The DoC's White Paper envisaged a "global, consensus, non-profit corporation", to serve as the means by which DNS management could be privatized, enabling an open, competitive system.

The ICANN Evolution and Reform exercise in 2002 has prompted the RIRs to carefully consider the relationship between the RIRs and ICANN. It is recognized that, like any private corporate entity, ICANN's continued existence is not protected, and ICANN's ability to continue to be in a position to execute all current IANA contracts should also not be simply assumed.

In approaching this issue the RIRs have undertaken some risk assessment in terms of the various external events that may compromise the RIRs ability to execute their role, or events that may compromise the stability of the Internet itself, and have conducted talks both with ICANN, and with a number of other stakeholders in the regulatory and public sectors.

The current position of the RIRs with respect to ICANN can be summarized as follows:

Policy Development. The RIRs believe that within the area of address management there is a valid role for a lightweight external review body with respect to global RIR policies, as part of an overall RIR requirement for check, balance and review in the global RIR policy determination process. The RIRs view this as a requirement for the policy development process to be protected to so that policy can only be made, changed, or overturned in a bottom-up process through open, transparent, and documented procedures.

Internet Number Resources. The RIRs believe that within the area of the Internet Number Resource pool that these resources are public resources. In this regard there is a valid role for a lightweight external body to coordinate the allocation of Internet Number Resources to the RIRs in accordance with global policies pertaining to the management of these resources and to protect the unallocated pool of these resources to which the RIRs must have free access, through established procedures at all times. This role may be carried out by ICANN, or by another suitably-qualified organisation.

REQUEST FOR COMMENTS (RFCS)

Documentation of work on the Internet, proposals for new or revised protocols, and TCP/IP protocol standards all appear in a series of technical reports called Internet Request for Comments, or RFCs. Preliminary versions of RFCs are known as Internet drafts. RFCs can be short or long, can cover broad concepts or details, and can be standards or merely proposals for new protocols. The RFC editor is a member of the IAB.

The RFC series is numbered sequentially in the chronological order RFCs are written. Each new or revised RFC is assigned a new number, so readers must be careful to obtain the highest numbered version of a document. Copies of RFCs are available from many sources including the IETF web page (www.ietf.org/rfc.html).

A unique standard (STD) number is assigned to each protocol reaching the maturity level of standard. The STD number identifies one or more RFCs that provide a specification for the protocol.

Although the RFC identified by the STD number may change, the STD number is constant. When a new RFC replaces an existing RFC, the existing RFC becomes obsolete. The replaced RFC number (or numbers) are listed under the title of "obsoletes" on the front page of the new RFC.

STANDARDS TRACK

Each RFC providing a specification of a protocol is assigned a "maturity level" (state of standardisation) and a "requirement level" (status). The maturity level of a Standards Track protocol begins with “proposed” standard. There are also protocols with a maturity level of “experimental”. Experimental protocols may remain experimental indefinitely, become "historic" or be reassigned as a "proposed standard" and enter the standards track.

Protocols with a proposed standard maturity level must be implemented and reviewed for a minimum of six months before progressing to "draft standard".

Progressing from draft standard to standard requires evaluation for a minimum of four months and the approval of the IESG and the IAB. The **diagram** illustrates the Standards Track process.

All RFCs can be obtained from the following web address, which also contains useful search tools, (www.rfc-editor.org/rfc.html).

STD number 1 lists the official protocol standards and Best Current Practice RFCs; however it is not a complete index to the RFC series. The STD indicates the most recent RFC relating to the official protocol standards. This RFC describes the state of standardisation of all protocols used in the Internet as determined by the IETF.

Each protocol has one of the following states of standardisation: standard, draft standard, proposed standard, experimental, informational or historic. Additionally each protocol has a requirement level: required, recommended, elective, limited use or not recommended.

See http://www.rfc-editor.org/rfc.html for further information on RFCs

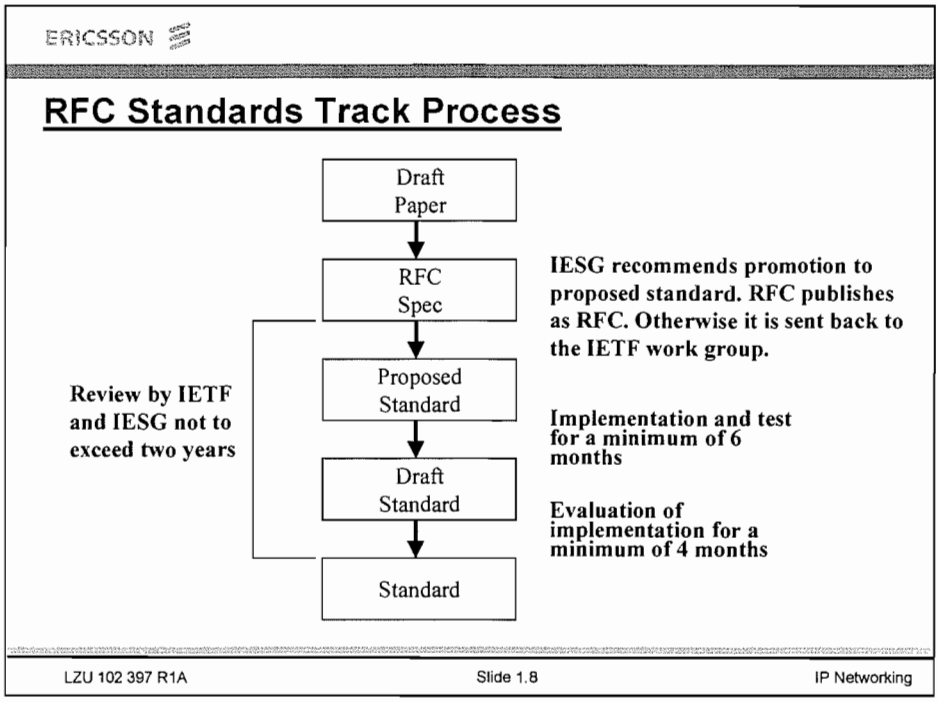


Figure 1-7.

Notes:

OSI 7-LAYER MODEL

The International Standards Organisation (ISO) has produced a protocol standard known as the Open Systems Interconnection (OSI) Reference Model. This consists of 7-layers that describe the hierarchical operation of specific functions in the communications process. Although the protocol itself has not gained wide acceptance, it is considered important as a means of identifying the factors and comparing the performance and capabilities of different protocols.

Each layer performs a well defined function, exchanging messages (relating to user data and control information) with the equivalent layer in another system, and having a well-defined interface to the layers immediately above and below itself.

The Physical Layer defines the type of medium, the transmission method, and the transmission rates available for the network.

The Data Link Layer defines how the network medium is accessed: which protocols are used, the packet / framing methods, and the virtual circuit / connection services.

The Network Layer standardises the way in which addressing is accomplished between linked networks.

The Transport Layer handles the task of reliable message delivery and flow control between applications on different devices.

The Session Layer establishes two-way communication between applications running on different devices on the network.

The Presentation layer translates data formats, so that devices with different "languages" can communicate

The Application Layer interfaces directly with the application programs running on the devices. It provides services such as file access and transfer peer-to-peer communication among applications, and resource sharing.

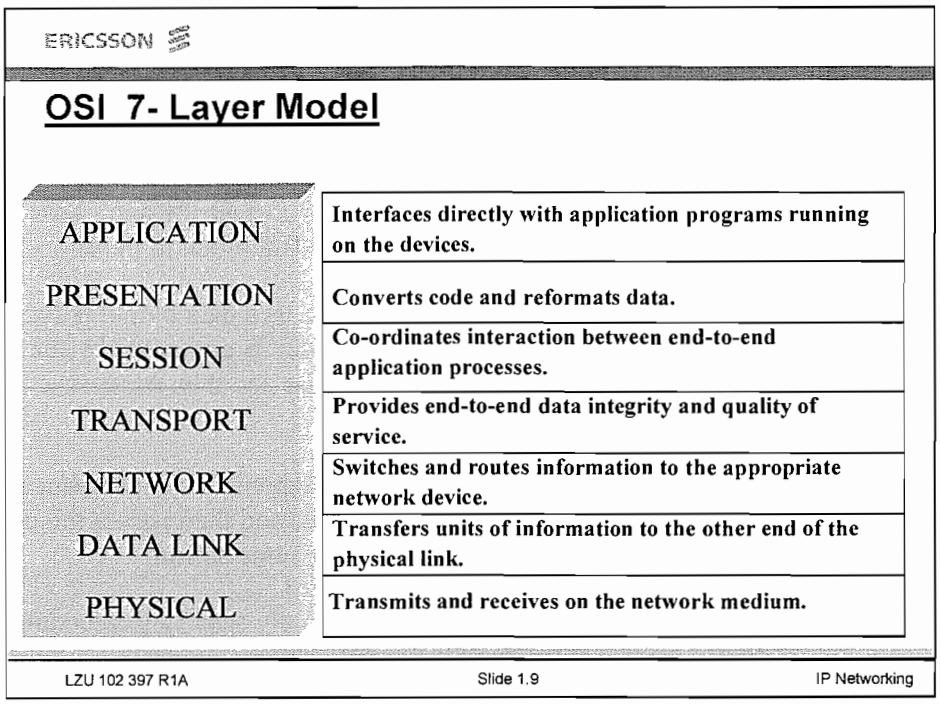


Figure 1-8.

Notes:

OSI 7-LAYER MODEL AND INTERNETWORKING DEVICES

Each layer operates independently of the others using a method referred to as encapsulation. At the sending device, each layer receiving data from the layer above processes the data, adds its own protocol header and transfer the data block to the layer below.

The layer below simply treats the data as a data block; it does not try to understand its meaning. The block is processed by the layer, which may add its own protocol header and then passes the larger data block to the next layer below.

At the receiving device the reverse happens. When the data arrives, the first layer processes its peer header and then passes the data to the layer above, which carries out the same action.

Ultimately, the application data originally sent by the sending device arrives at the receiving application.

Routers operate at the network layer. They connect networks into inter-networks that are physically unified, but in which each network retains its identity as a separate network environment.

Bridges and Layer 2 switches operate at the Data link layer. They connect network environments into logical and physical single inter-networks.

Repeaters operate at the Physical layer. They receive transmissions (bits) on a LAN segment and regenerate the bits to boost a degraded signal and extend the length of the LAN segment.

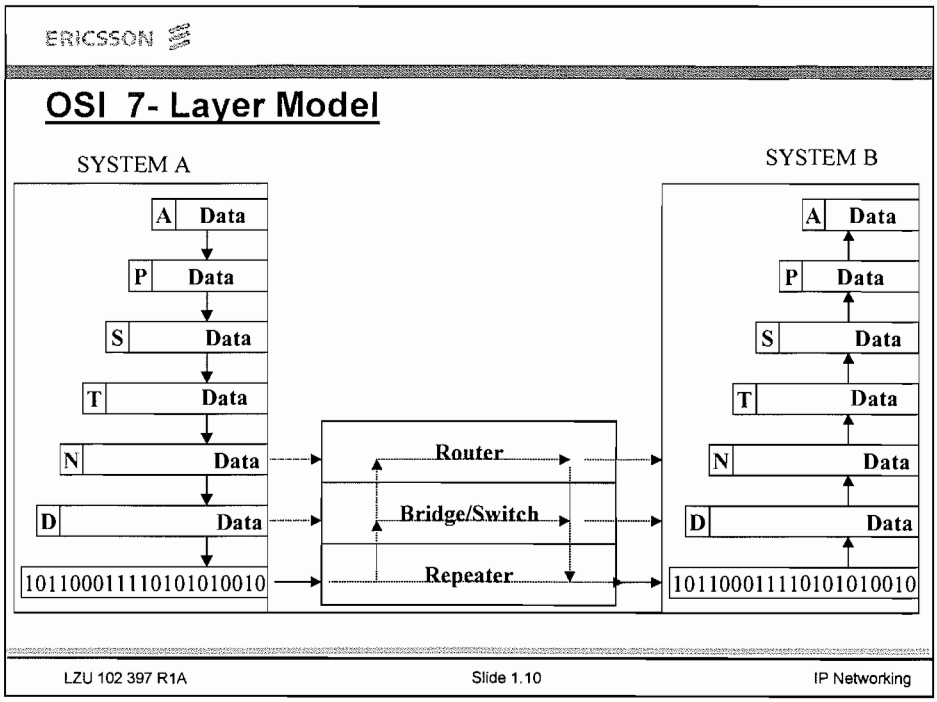


Figure 1-9.

Notes:

TCP/IP SUITE

Transmission Control Protocol/Internet Protocol (TCP/IP) is not a single protocol; it refers to a family or suite of protocols. The suite consists of a four-layer model.

Network Interface Layer

The Network Interface Layer is equivalent to the combination of the Physical and Data Link Layers in the OSI model. It is responsible for formatting packets and placing them onto the underlying network. All common Data Link protocols support TCP/IP.

Internet Layer

The Internet Layer is equivalent to the Network Layer in the OSI model. It is responsible for network addressing. The main protocols at this layer are: Internet Protocol (IP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Internet Control Message Protocol (ICMP), and Internet Group Management Protocol (IGMP).

The Transport Layer

The Transport Layer is equivalent to the Transport Layer in the OSI model. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) implement the Internet Transport layer. TCP provides reliable data transport, while UDP provides unreliable data transport.

The Application Layer

The Application Layer is equivalent to the top three layers, (Application, Presentation and Session Layers), in the OSI model. The Application Layer is responsible for interfacing between user applications and the Transport Layer. Applications commonly used are: Domain Name system (DNS), File Transfer Protocol (FTP), Telnet, Simple Network Management Protocol (SNMP), Simple Mail Transfer Protocol (SMTP), and so on.

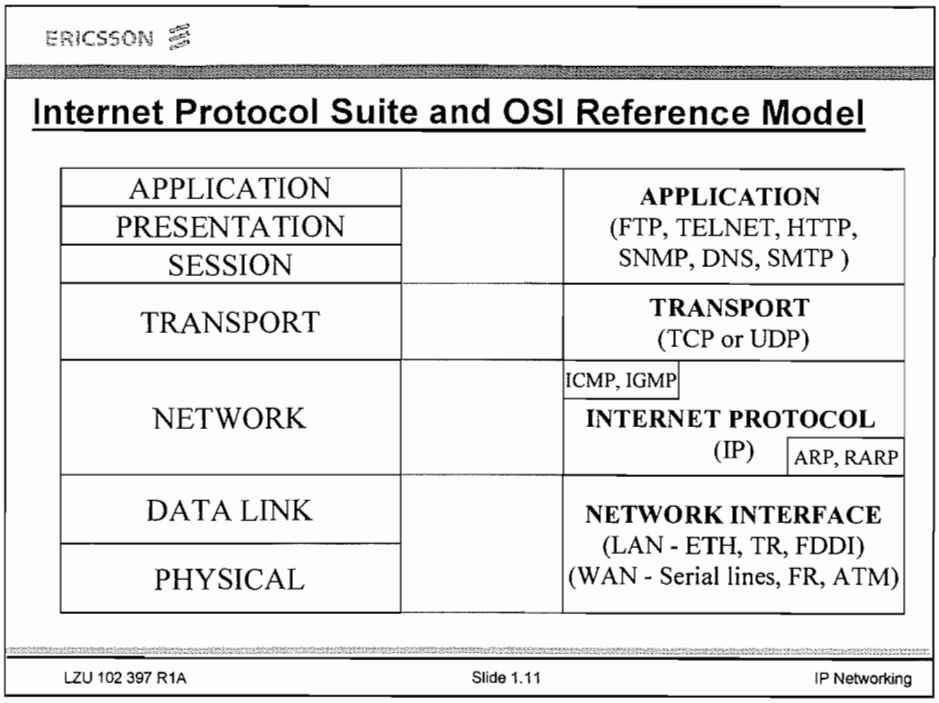


Figure 1-10.

Notes:

TCP / IP Protocol stack based on Data flow

In reality, the interaction between the various protocols is more complex than illustrated in the previous diagram.

For example, the Internet Control Message Protocol (ICMP) and the Internet Group Message Protocol (IGMP) are an integral part of the Internet layer. However, each receives data and control in the same manner as a Transport layer function, namely, by an assigned protocol number contained in the IP header.

Hence, they are illustrated in this diagram of the TCP/IP protocol stack based on data flow and control. For the same reason, some other protocols may be identified in Internet literature differently than in this illustration.

For example, the Routing Information Protocol (RIP) has an assigned port number, contained in a User Datagram Protocol (UDP) header, making it an upper layer protocol.

Yet, another routing protocol, Open Shortest Path First (OSPF) has an assigned protocol number, making it a transport layer protocol. Similarly the Border Gateway Protocol (BGP) uses a port number from the TCP header for data flow and control.

In theory, all upper layer protocols could use either UDP or TCP. Both provide a transport layer function. The reliability requirements of the applications dictate, which transport layer, is used. UDP provides an unreliable, connectionless transport service, while TCP provides a reliable, in sequence, connection oriented service.

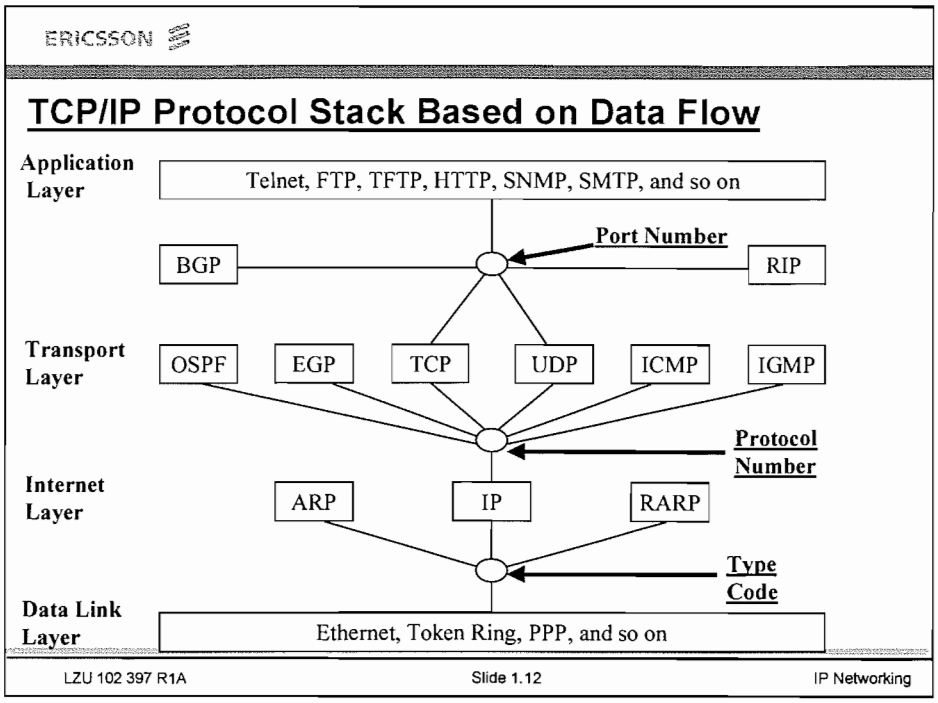


Figure 1-11.

Notes:

Intentionally Blank

2 IP Addressing

After completing this chapter you will be able to:

- Identify the characteristics and features of IP
- Describe IPv4 addressing
- Describe IPv6 addressing

Intentionally Blank

INTERNET PROTOCOL (IP)

IP is a connectionless protocol that is primarily responsible for addressing and routing packets between network devices. Connectionless means that a session is not established before data is exchanged. The internet protocol provides for transmitting blocks of data called 'datagrams' from sources to destinations, where sources and destinations are hosts identified by fixed length addresses.

IP is unreliable because packet delivery is not guaranteed. IP makes what is termed a 'best effort' attempt to deliver a packet. Along the way a packet may be lost, delivered out of sequence, duplicated or delayed. An acknowledgement is not required when data is received. The sender or receiver is not informed when a packet is lost or out of sequence. The acknowledgement of packets is the responsibility of a higher-layer transport protocol, such as the Transmission Control Protocol (TCP).

The Internet protocol also provides for fragmentation and reassembly of long datagrams, if necessary, for transmission through "small packet" networks. A large datagram must be divided into smaller pieces when it has to traverse a network that supports a smaller packet size. For example, an IP packet on a Fiber Distributed Data Interface (FDDI) network may be up to 8,968 bytes long. If such a packet needs to traverse an Ethernet network, it must be split up into IP packets, which are a maximum of 1500 bytes long.

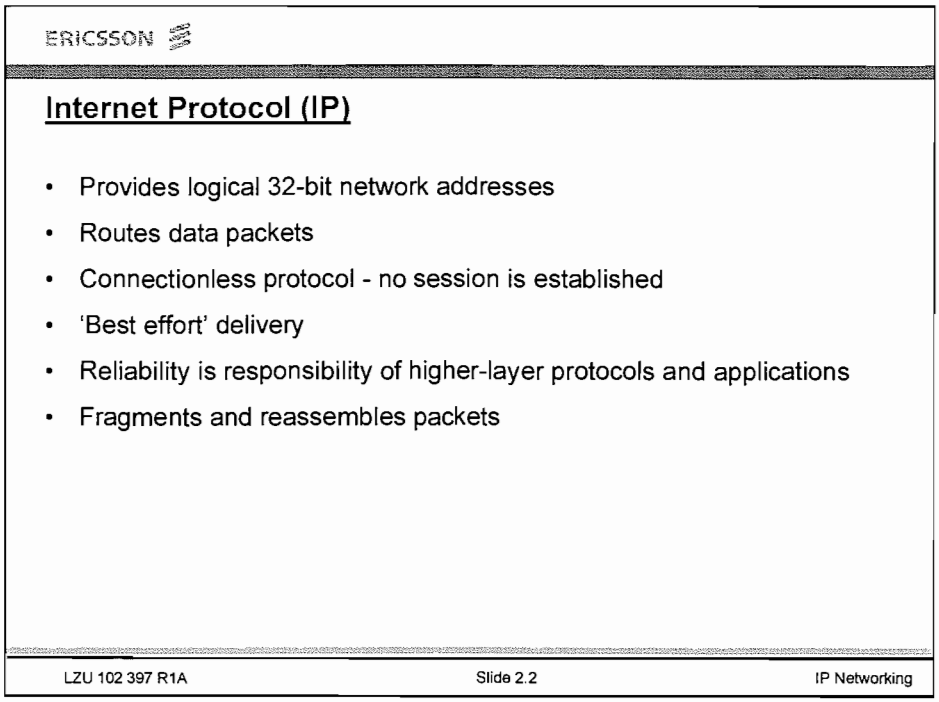


Figure 2-1.

Notes:

Routing of IP Packets

The Internet modules use the addresses carried in the internet header to transmit internet datagrams toward their destinations. The selection of a path for transmission is called routing. IP delivers its packets in a connectionless mode. It does not check to see if the receiving host can accept data. Furthermore it does not keep a copy in case of errors or retransmission. IP is therefore said to "fire and forget".

When a packet arrives at a router, the router forwards the packet only if it knows a route to the destination. If it does not know the destination, it drops the packet. In practice routers rarely drop packets, because they typically have default routes defined.

IP may silently discard a packet in some error situations, with or without notification to the originator. The method used to notify the originator is the Internet Control Message Protocol (ICMP). This protocol is an integral part of IP.

The router does not send any acknowledgements to the sending device. A router analyses the checksum. If it is not correct then the packet is dropped. It also decreases the Time-To-Live (TTL), and if this value is zero, then the packet is dropped and an ICMP message is sent to the originator.

If necessary the router fragments larger packets into smaller ones and sets flags and Fragment Offset fields accordingly. Finally, a new checksum is generated due to changes in TTL and possibly flags and Fragment Offset. The packet is then forwarded.

The specifications for IP and ICMP are contained in RFC 791 and RFC 792, respectively. (Defined in IETF Standard #5.)

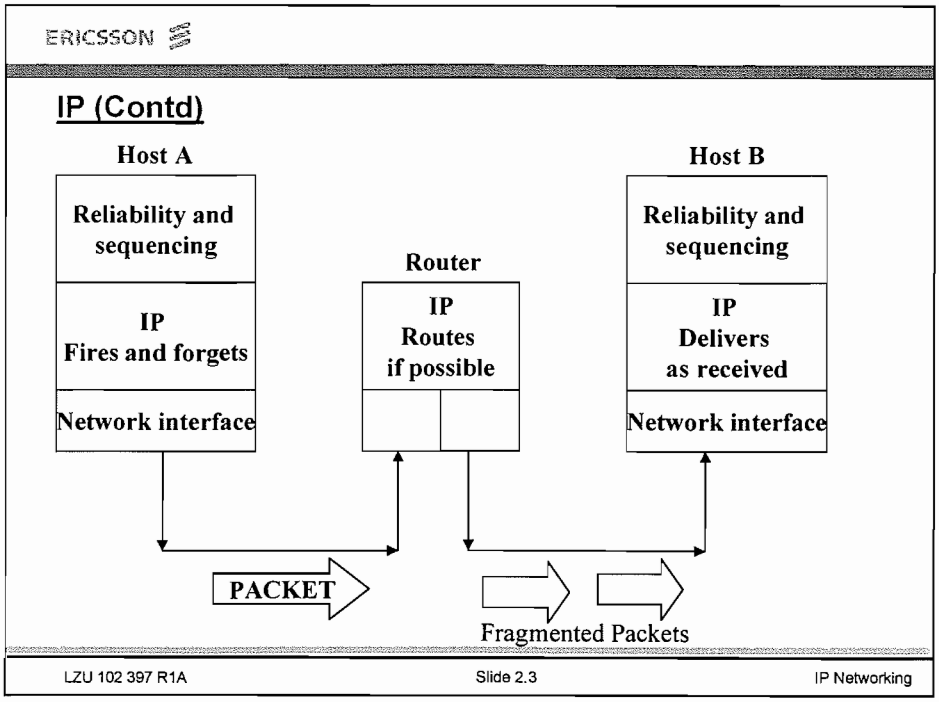


Figure 2-2.

Notes:

IP PACKET FORMAT

The diagram shows the format of an IP packet and it's constituent components

Version

Although the range of values is 0 to 15, the value used by IP version 4 (IPv4) is equal to 4. By means of this field, different versions of the IP could operate in the Internet. When using IPv6 the value of this field is equal to 6.

Internet Header Length (IHL)

Indicates the length of the header in 32-bit words. The minimum value is five, which is the most common header. The header must be at least 20 bytes in length. The IHL is used as a pointer to the beginning of data. When options are used, padding may be required to make the total size if the header an even multiple of 32-bit words. The value of the header length may range from 5 to 15. The Options provide for control functions needed or useful in some situations but unnecessary for the most common communications. The options include provisions for timestamps, security, and special routing.

Type of Service (TOS)

The Type of Service field specifies the precedence and priority of the IP datagram. Traditionally, the Type of Service field was unused. In recent years an IETF technology called "Diffserv" has been developed which makes use of this field.

Total Length

The total length field is used to identify the number of octets in the entire datagram. The field has 16 bits and the range is between 0 and 2^{16}-1 (65,535) octets. Since the datagram is typically contained in an Ethernet frame, the size will usually be equal or less than 1,500 octets. Larger datagrams may be handled by some intermediate networks of the Internet, but are segmented if a network router is unable to handle the larger size. The IPv4 specification sets a minimum size of 576 octets that must be handled by routers without fragmentation. Datagrams larger than this may be subject to fragmentation.

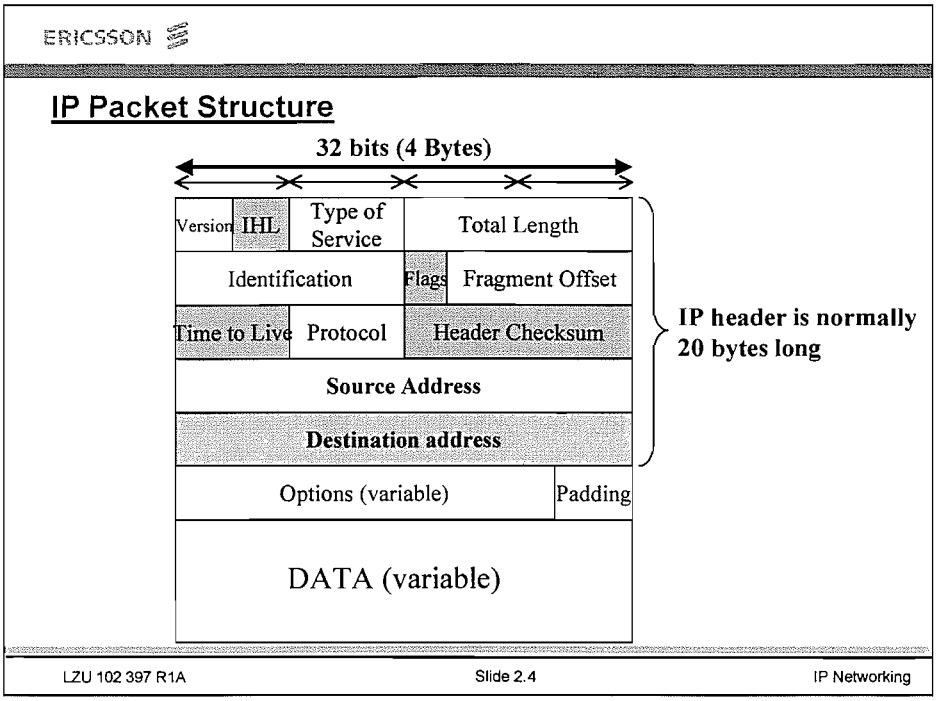


Figure 2-3.

Notes:

Identification

The value of the 16-bit identification field is a sequential number assigned by the originating host to aid in reassembling a fragmented datagram. The primary purpose is to allow the destination device to collect all fragments from a datagram, since they will all have the same identification number.

Flags

The flag field contains two flags. The low order bit is used to denote the last fragment when set to zero. That is, intermediate (non-last) fragments have the bit set equal to one to denote more fragments are to follow and it is set to zero in the last fragment. In non-fragmented datagrams this bit is set to zero. The second bit is set by an originating host to prevent fragmentation of the packet. When this bit is set to one and the length of the datagram exceeds that of an intermediate network, the intermediate network discards the datagram and an error message is returned to the originating host via the ICMP. The third bit is not used and is set to zero.

Fragment Offset

When the size of a datagram exceeds the maximum of an intermediate network, that network segments it. The 13-bit fragment offset field represents the displacement of this segment from the beginning of the original datagram. Since the value represents groups of eight octets, the effective range of the offset is between zero and 65,535 ($(2^{13} * 2^3)-1$)) octets. The resulting fragments are treated as complete datagrams and remain that way until they reach the destination host where they are reassembled into the original datagram. The fragment-offset field is used to assemble the fragmented datagrams as they may arrive out of sequence.

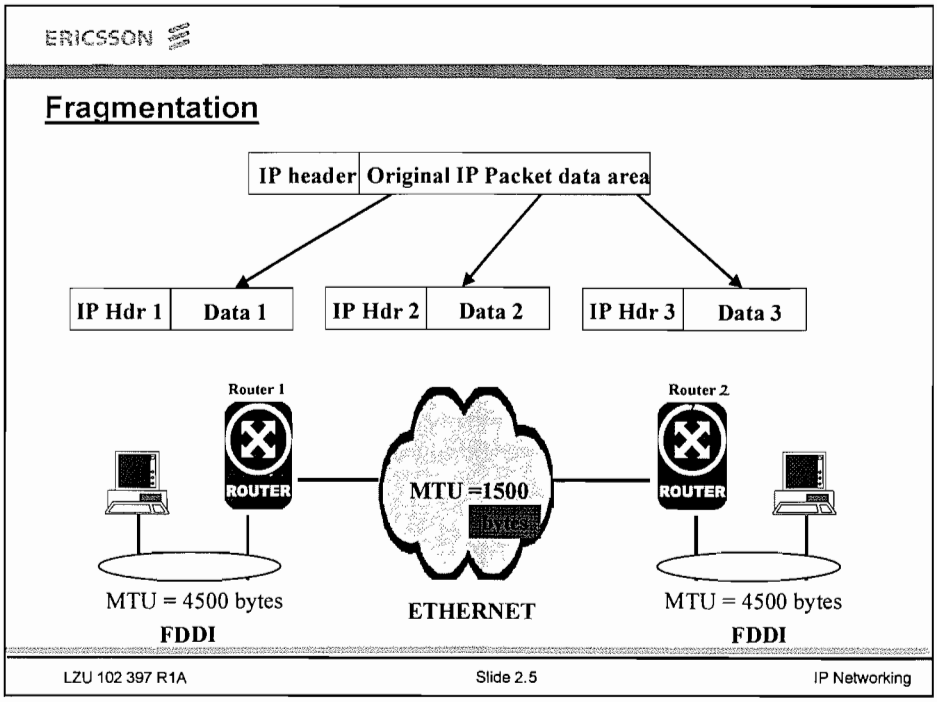


Figure 2-4.

Notes:

Time to Live

The Time To Live (TTL) field represents a count (in seconds) set by the originator that the datagram can exist in the Internet before being discarded. Hence, a datagram may loop around an Internet for a maximum of 255 (2^8-1) seconds before being discarded. The current recommended default TTL for the IP is 64. Since each router handling a datagram decrements the TTL by a minimum of one, the TTL can also represent a hop count. However, if the router holds the datagram more than one second, it decrements the TTL by the number of seconds held. The originator of the datagram is sent an error message via the ICMP when the packet is discarded.

Protocol

The protocol field is used to identify the next higher layer protocol using the IP. It will normally identify either the TCP (equal to 6 decimal) or UDP (equal to 17 decimal) Transport Layer, but may identify up to 255 different Transport Layer protocols. An upper layer protocol using IP must have a unique protocol number.

See http://www.iana.org/assignments/protocol-numbers

Checksum

The 16-bit checksum field provides assurance that the header has not been corrupted during transmission. The checksum includes all fields in the IP header, starting with the version number and ending with the octet immediately preceding the IP data field, which may be a pad field if the option field is present. The checksum includes the checksum field itself, which is set to zero for the calculation. The checksum represents the 16-bit one's complement of the one's complement sum of all 16-bit groups (double octet pairs) in the header. An intermediate router that changes a field in the IP header (e.g., time-to-live) must recalculate the checksum before forwarding it. Users of IP must provide their own data integrity, since the IP checksum is only for the header.

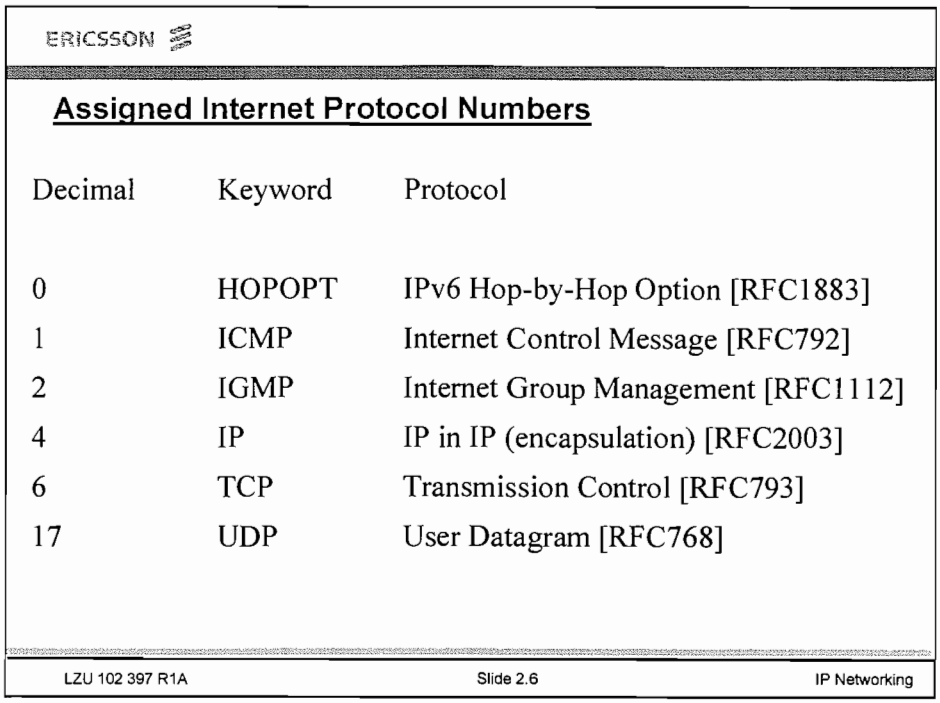


Figure 2-5.

Notes:

THE IPV4 ADDRESS

The 32-bit source and destination IP address fields contains the network and host identifiers of the originating and destination end points, respectively.

Every network interface on a TCP/IP device is identified by a globally unique IP address. Host devices, for example, PCs, typically have a single IP address. Routers typically have two or more IP addresses, depending on the number of interfaces they have. Each IPv4 address is 32 bits long and is composed of four 8-bit fields called octets. The address is normally represented in 'dotted decimal notation' by grouping the four octets and representing each one in decimal form. A decimal number in the range 0-255 then represents each octet.

For example, 11000001 10100000 00000001 00000101, is represented as 193.160.1.5.

Each IP address consists of a network ID and a host ID. The network ID identifies the systems that are located on the same network. The network ID must be unique to the internetwork. The host ID identifies a TCP/IP network device (or host) within a network. The address for each host must be unique to the network ID. In the example above, the PC is connected to network '193.160.1.' and has a unique host ID of '.5'.

The Internet Assigned Numbers Authority (IANA) has ultimate control over network IDs assigned and sets the policy. The IANA has delegated this responsibility to four regional Internet registries:

ARIN (American Registry for Internet Numbers) (http://www.arin.net)

RIPE (Réseaux IP Européens Network Coordination Centre) (http://www.ripe.net)

APNIC (Asia Pacific Network Information Centre) (http://www.apnic.net)

LACNIC (Latin American and Caribbean Internet Addresses Registry (http://www.lacnic.net)

Internet service providers (ISPs) apply to their regional Internet registry to get blocks of addresses referred to as address space. The ISPs assign addresses from those address spaces to their customers, for example, companies that want to connect to the Internet.

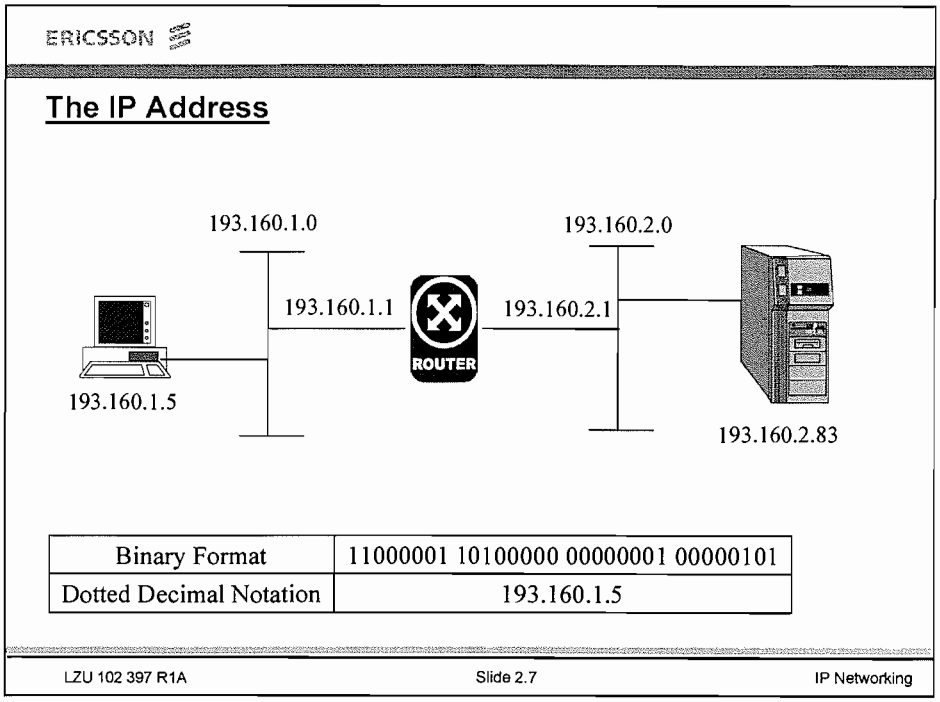


Figure 2-6.

Notes:

Binary to Decimal

Each bit position in an octet has an assigned decimal value. A bit set to zero always has a zero value. The lowest order bit has a decimal value of 1. The highest order bit has a decimal value of 128. The highest decimal value of an octet is 255, that is, when all bits are set to one. In the example below, the binary value 10011000 is converted to a decimal value of 152.

Binary Value	1	0	0	1	1	0	0	0
	2^7	0	0	2^4	2^3	0	0	0
Decimal Value	128	0	0	16	8	0	0	0

The binary value 10011000 is 152 decimal. This is 128+16+8=152.

Note that occasionally IP addresses are written in hexadecimal notation. In order to convert from binary to hexadecimal, take each block of four bits and change to the hexadecimal equivalent, for example, 1001 1000 is equal to 98 in hex.

Example:

163.33.232.166 = 10100011.00100001.11101000.10100110 = A3.21.E8.A6 (Hex)

ERICSSON

Converting from Binary to Decimal

Binary Value	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Value	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**

If all bits are set to 1 then the decimal value is 255, that is,
1+2+4+8+16+32+64+128=255

LZU 102 397 R1A Slide 2.8 IP Networking

Figure 2-7.

Notes:

TRADITIONAL IP ADDRESS CLASSES

The first part of an Internet address identifies the network, on which a host resides, while the second part identifies the particular host on a given network. The network-ID field can also be referred to as the network-number or the network-prefix. All hosts on a given network share the same network-prefix but must have a unique host-number. There are five different address classes supported by IP addressing. The class of an IP address can be determined from the high-order (left-most) bits.

Class A (/8 Prefixes)

Class A addresses were assigned to networks with a very large number of hosts. The high-order bit in a class A address is always set to zero. The next seven bits (completing the first octet) represent the network ID and provide 126 possible networks. The remaining 24 bits (the last three octets) represent the host ID. Each network can have up to 16,777,214 hosts.

Class B (/16 Prefixes)

Class B addresses were assigned to medium-sized to large-sized networks. The two high-order bits in a class B address are always set to binary 1 0. The next 14 bits (completing the first two octets) represent the network ID. The remaining 16 bits (last two octets) represent the host ID. Therefore, there can be 16382 networks and up to 65534 hosts per network.

Class C (/24 Prefixes)

Class C addresses were used for small networks. The three high-order bits in a class C address are always set to binary 1 1 0. The next 21 bits (completing the first three octets) represent the network ID. The remaining 8 bits (last octet) represent the host ID. There can, therefore, be 2097150 networks and 254 hosts per network.

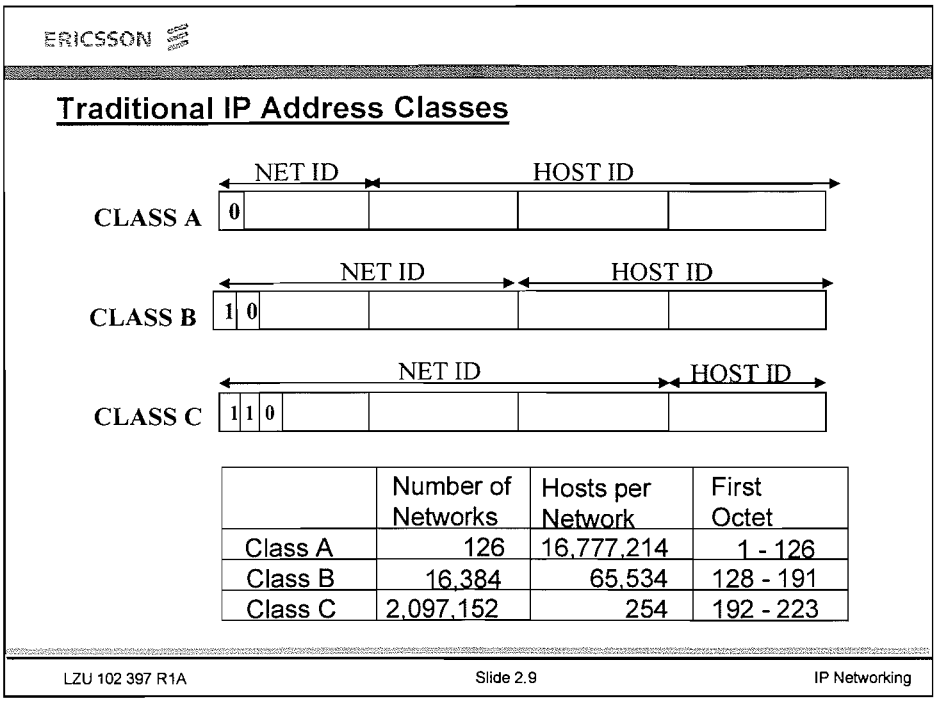


Figure 2-8.

Notes:

MULTICAST & RESERVED

Class D

Class D addresses are employed for multicast group usage. A multicast group may contain one or more hosts or none at all.

The four high-order bits in a class D address are always set to binary 1 1 1 0. The remaining bits designate the specific group, in which the client participates.

When expressed in dotted decimal notation, multicast addresses range from 224.0.0.0 through 239.255.255.255. There are no network or host bits in the multicast operations. Packets are passed to a selected subset of hosts on a network. Only those hosts registered for the multicast operation accept the packet. Some multicast group addresses are assigned as well known addresses by the IANA. For example, the multicast address 224.0.0.6 is used for OSPF hello messages, and 224.0.0.9 is used for RIP-2.

Class E

Class E is an experimental address not available for general use. It is reserved for future use. The high-order bits in a class E address are set to 1 1 1 1

Extract from RFC1812 "Requirements for IPv4 Routers"

'The explosive growth of the Internet has forced a review of address assignment policies. The traditional uses of general-purpose (Class A, B, and C) networks have been modified to achieve better use of IP's 32-bit address space. Classless Inter Domain Routing (CIDR) is a method currently being deployed in the Internet backbones to achieve this added efficiency. CIDR depends on deploying and routing to arbitrarily sized networks. In this model, hosts and routers make no assumptions about the use of addressing in the Internet. The Class D (IP Multicast) and Class E (Experimental) address spaces are preserved, although this is primarily an assignment policy.'

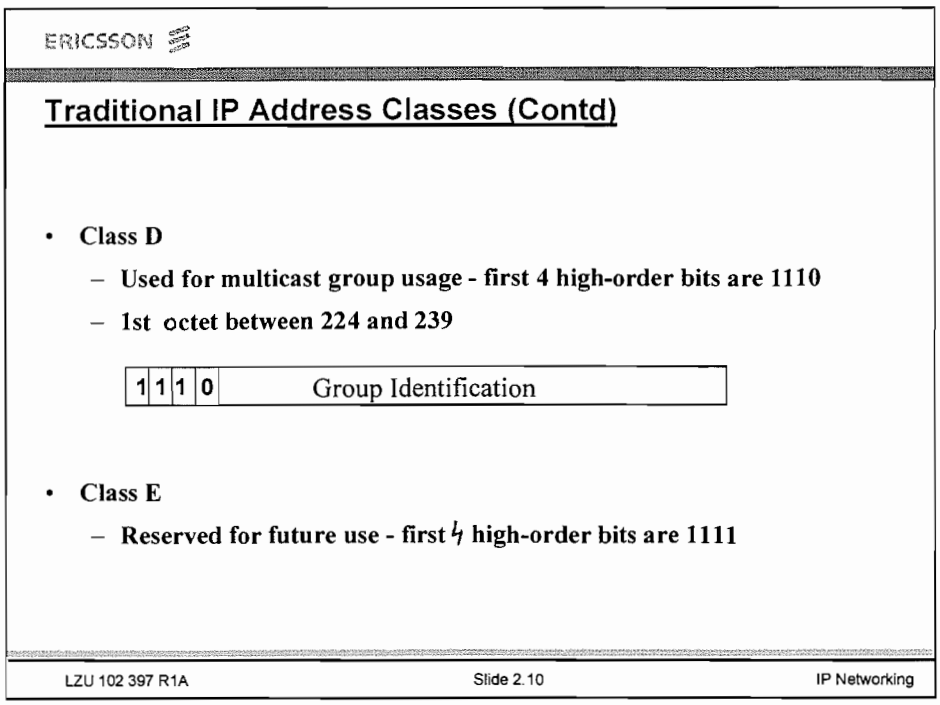


Figure 2-9.

Notes:

Guidelines

The following rules must be adhered to when assigning network IDs and host IDs:

- The network ID cannot be 127. The class A network address 127.0.0.0 is reserved for loop-back and is designed for testing and inter-process communication on the local device. When any device uses the loop-back address to send data, the protocol software in the device returns the data without sending traffic across any network.
- The network ID and host ID bits of a specific device cannot be all 1s. If all bits are set to 1, the address is interpreted as a broadcast rather than a host ID. The following are the two types of broadcast:
- If a destination address contains all 1s in the network ID and the host ID (255.255.255.255) then it, is a limited broadcast, that is, a broadcast on the source's local network.
- If a destination address contains all 1s in the host ID but a proper network ID, for example, 160.30.255.255, this is a directed broadcast, that is, a broadcast on a specified network (in this example network 160.30.0.0)
- The network ID and host ID bits cannot all be 0s. If all bits are set to 0, the address is interpreted to mean 'this network only'.
- The host ID must be unique to the local network.

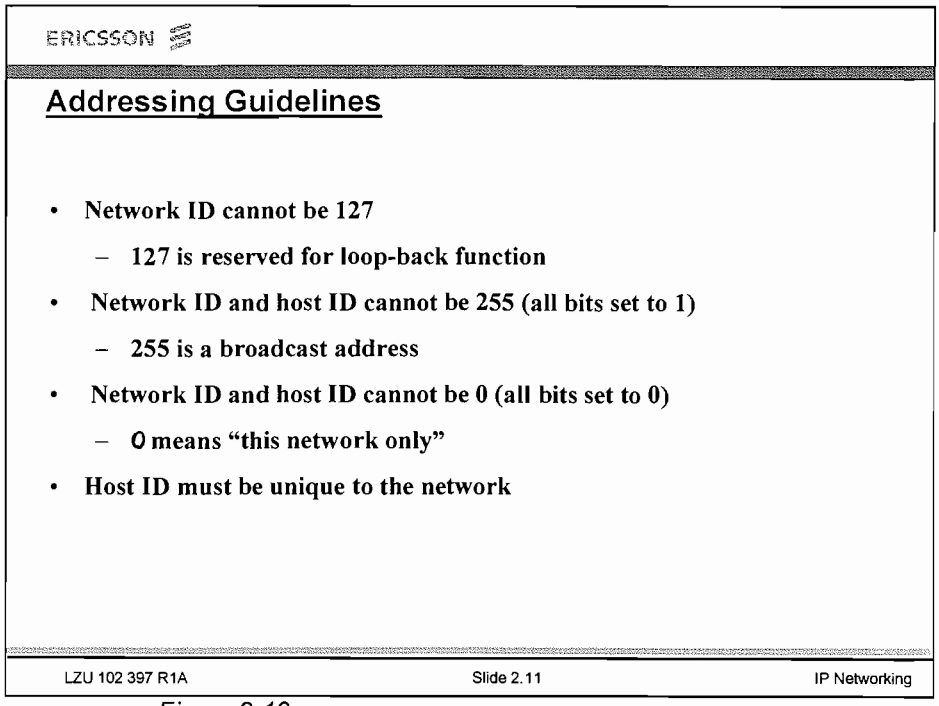


Figure 2-10.

Notes:

PRIVATE IP ADDRESS SPACE

RFC 1918 requests that organisations make use of the private Internet address space for hosts which require IP connectivity within the enterprise network, but do not require external connections to the global Internet. For this purpose the IANA has reserved the following three address blocks for private Internets:

10.0.0.0 - 10.255.255.255

172.16.0.0 - 172.31.255.255

192.168.0.0 - 192.168.255.255

Any organisation that elects to use addresses from these reserved blocks can do so without contacting the IANA or an Internet registry. Since these addresses are never injected into the global Internet routing system, the address space can be used simultaneously by many organisations. The disadvantage of this addressing scheme is that it requires an organisation to use a Network Address Translator (NAT) for global Internet access.

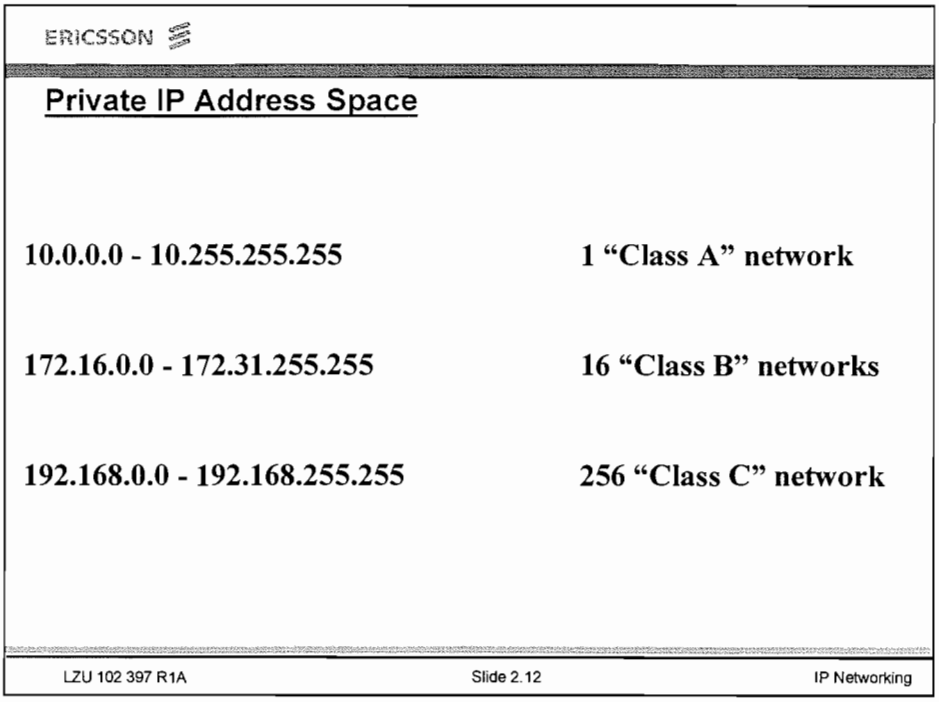


Figure 2-11.

Notes:

Special-Use IPv4 Addresses [RFC 3330]:

This RFC describes the global and other specialized IPv4 address blocks that have been assigned by the Internet Assigned Numbers Authority (IANA).

Address Block	**Present Use**
0.0.0.0/8	"This" Network
10.0.0.0/8	Private-Use Networks
14.0.0.0/8	Public-Data Networks
24.0.0.0/8	Cable Television Networks
39.0.0.0/8	Reserved but subject to allocation
127.0.0.0/8	Loopback
128.0.0.0/16	Reserved but subject to allocation
169.254.0.0/16	Link Local
172.16.0.0/12	Private-Use Networks
191.255.0.0/16	Reserved but subject to allocation
192.0.0.0/24	Reserved but subject to allocation
192.0.2.0/24	Test-Net
192.88.99.0/24	6to4 Relay Anycast
192.168.0.0/16	Private-Use Networks
198.18.0.0/15	Network Interconnect Device Benchmark Testing
223.255.255.0/24	Reserved but subject to allocation
224.0.0.0/4	Multicast
240.0.0.0/4	Reserved for Future Use

ERICSSON

Special Use IP Address Space

0.0.0.0/8	"This" Network
14.0.0.0/8	Public-Data Networks
24.0.0.0/8	Cable Television Networks
39.0.0.0/8	Reserved but subject to allocation
127.0.0.0/8	Loopback
128.0.0.0/16	Reserved but subject to allocation
169.254.0.0/16	Link Local
192.0.0.0/24	Reserved but subject to allocation
192.0.2.0/24	Test-Net
192.88.99.0/24	6to4 Relay Anycast
223.255.255.0/24	Reserved but subject to allocation

LZU 102 397 R1A Slide 2.13 IP Networking

Figure 2-12.

Notes:

SUBNET MASK

A subnet mask is a 32-bit address used to:

- Block out a portion of the IP address to distinguish the network ID from the host ID.
- Specify whether the destination host's IP address is located on a local network or on a remote network.

For example, an IP device with the IP Address 160.30.20.10 and Subnet Mask 255.255.255.0 knows that its network ID is 160.30.20 and its host ID is .10

For convenience the subnet mask can be written in prefix length notation. The prefix-length is equal to the number of contiguous one-bits in the subnet mask. Therefore, the network address 160.30.20.10 with a subnet mask 255.255.255.0 can also be expressed as 160.30.20.10/24.

(255.255.255.0 = 11111111.11111111.11111111.00000000 in binary)

The default subnet masks or prefix lengths for a Class A address is 255.0.0.0 or /8, Class B default mask 255.255.0.0 or /16 and a Class C default mask 255.255.255.0 or /24

ANDing is an internal process that TCP/IP uses to determine whether a packet is destined for a host on a local network, or a host on a remote network. When TCP/IP is initialised, the host's IP address is ANDed with its subnet mask. Before a packet is sent, the destination IP address is ANDed with the same subnet mask. If both results match, IP knows that the packet belongs to a host on the local network. If the results don't match, the packet is sent to the IP address of an IP router. To AND the IP address to a subnet mask, TCP/IP compares each bit in the IP address to the corresponding bit in the subnet mask. If both bits are 1s, the resulting bit is 1. If there is any other combination, the resulting bit is 0. The four possible variations are as follows:

1	AND	1	1
1	AND	0	0
0	AND	1	0
0	AND	0	0

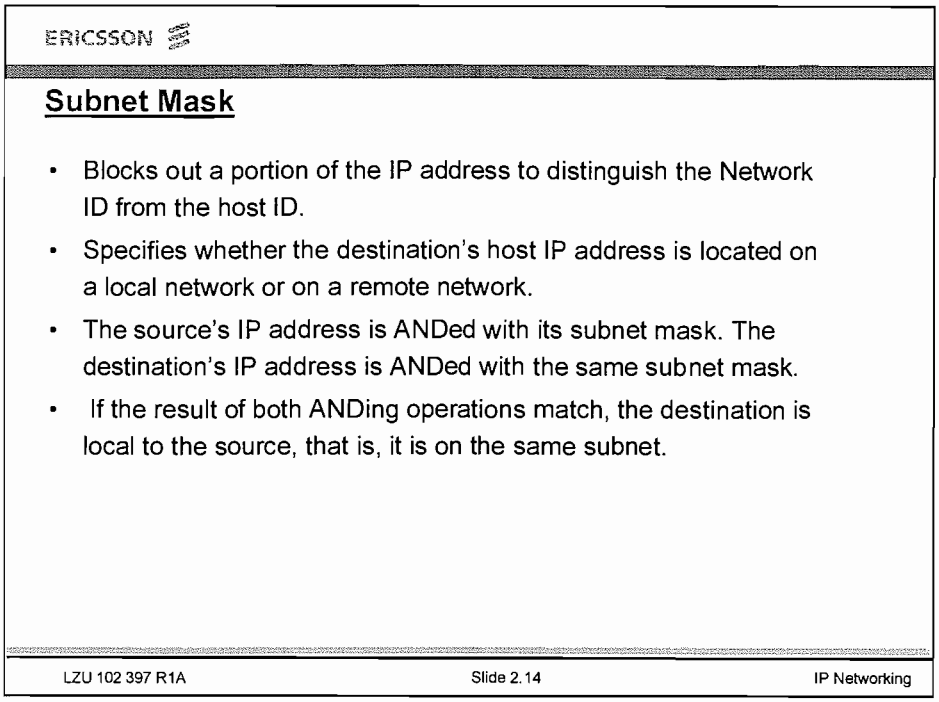


Figure 2-13.

Notes:

Subnetting

Subnetting was initially introduced to overcome some of the problems that parts of the Internet were beginning to experience:

- Internet routing tables were becoming too large to manage.
- Local administrators had to request another network number from the Internet before a new network could be installed at their site.

Subnetting attacked the expanding routing table problem by ensuring that the subnet structure of a network is never visible outside of the organisation's private network. The route from the Internet to any subnet of a given IP address is the same, regardless of which subnet the destination host is on. This is because all subnets of a given network ID use the same network prefix, but different subnet numbers. The routers within the private organisation need to differentiate between the individual subnets, but as far as the Internet routers are concerned all of the subnets in the organisation are collected into a single routing table entry.

Subnetting helps to overcome the registered number issue by assigning each organisation one (or in some cases a few) network number(s) from the IPv4 address space. The organisation is then free to assign a distinct subnetwork number to each of its internal networks. This allows the organisation to deploy additional subnets without needing to obtain a new network number from the Internet.

For example, a site with several logical networks uses subnet addressing to cover them with a single 'class B' network address. The router accepts all traffic from the Internet addresses to network 160.30.0.0, and forwards traffic to the internal subnetworks based on the third octet of the classful address. The deployment of subnetting within the private network provides several benefits:

The size of the global Internet routing table does not grow because the site administrator does not need to obtain additional address space, and the routing advertisements for all of the subnets are combined into a single routing table entry.

The local administrator has the flexibility to deploy additional subnets without obtaining a new network number from the Internet. Rapid changing of routes within the private network does not affect the Internet routing table, since Internet routers do not know about the reachability of the individual subnets. They just know about the reachability of the parent network number.

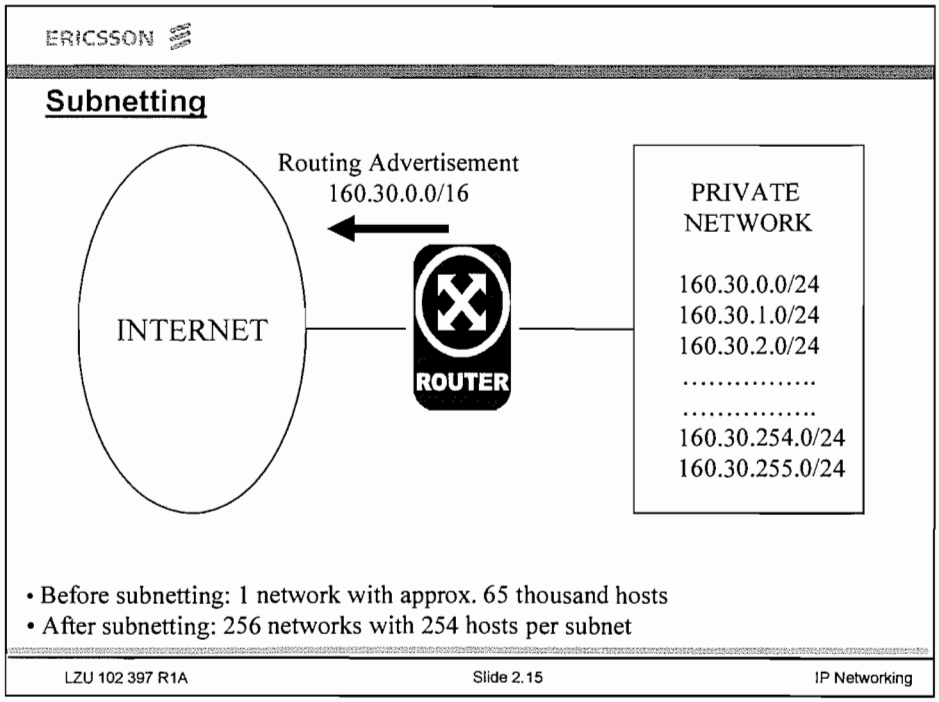


Figure 2-14.

Notes:

NETWORK WITH CUSTOMISED MASK

The example shown calculates the number of subnets available when a customised mask is applied.

The IP address space 160.30.0.0/16 has been allocated to an organisation.

Using the default subnet mask on a 'class B' network gives one single network with a total of 65534 hosts.

Using the customised mask 255.255.255.0, the organisation has up to 256 subnets, rather than just one single network.

A shortcut method of working out the number of subnets is: (2 to the power of the number of bits in the subnet mask, excluding the default mask portion). In the example above this is 2^8, which gives a total of 256 subnets.

Calculating the number of hosts per subnet

This is the same example, but this time we want to calculate the number of hosts in any one of the 256 subnets.

The host addresses 0 and 255 cannot be used. Therefore the lowest possible host address on each subnet is 1, and the highest possible host address on each subnet is 254.

A shortcut method of working out the number of hosts in a subnet is:

{(2 to the power of the number of zeros in the mask) minus two}. In the example shown this is 2^8 -2, which gives a total of 254 hosts per subnet.

ERICSSON

Example: Network with Customised Mask

Allocated IP address space 160.30.0.0/16

3-octet mask 255.255.255.0				
8 bits available for subnets and 8 bits available for host				
	255	255	255	0
	1111 1111	1111 1111	1111 1111	0000 0000
No. of Subnets	**Network**			**Host**
160.30.0.x	1010 0000	0001 1110	**0000 0000**	xxxx xxxx
↓			↓	
160.30.255.x	1010 0000	0001 1110	**1111 1111**	xxxx xxxx
Maximum of 256 subnets (2^8)				

LZU 102 397 R1A Slide 2.16 IP Networking

Figure 2-15.

Notes:

NETWORK WITH VLSM

Variable Length Subnet Masks (VLSM) support more efficient use of an organisation's assigned IP address space. One of the major problems with the earlier limitation of supporting only a single subnet mask across a given network-prefix was that once the mask was selected, it locked the organisation into a set number of fixed size subnets.

For example, assume that a network administrator decided to configure the 200.200.200.0/24 with a /26 extended-network-prefix (subnet mask). This permits four subnets, each of which supports a maximum of 62 devices.

Alternatively, if we configure with a /28 extended-network-prefix then this permits 16 subnets with 14 hosts each. Neither of these is suitable if we want 2 subnets with 50 hosts and 8 subnets with 10 hosts. If the /26 mask is used throughout the network then there are not enough subnets. If the /28 mask is used throughout the network then there are not enough host addresses for two of the subnets.

ERICSSON

Example: Network with Customised Mask

Allocated IP address space 160.30.0.0/16

3-octet mask 255.255.255.0				
8 bits available for subnets and 8 bits available for host				
	255	255	255	0
	1111 1111	1111 1111	1111 1111	0000 0000
No. of hosts	**Network**			**Host**
160.30.x.1	1010 0000	0001 1110	xxxx xxxx	**0000 0001**
↓				↓
160.30.x.254	1010 0000	0001 1110	xxxx xxxx	**1111 1110**
Maximum of 254 hosts (2^8 - 2)				

LZU 102 397 R1A Slide 2.17 IP Networking

Figure 2-16.

Notes:

VLSM (Cont)

The solution to this problem is VLSM, which allows a subnetted network to be assigned more than one subnet mask. In this example, VLSM allows us to use both a /26 mask and a /28 mask. We use the /26 mask to produce two subnets with a maximum of 62 devices each. We use the /28 mask to produce eight subnets with a maximum of 14 hosts each. This is suitable for our stated requirements.

The diagram shows an example of a portion of a real network with VLSM implemented.

The company owns the address block 160.40.0.0/16. On Site A all devices are on the same subnet. There can be a maximum of 1022 devices (2^{10} -2), since there are 10 bits available for host addresses. Valid network addresses are from 160.40.144.1 to 160.40.147.254.

Similarly, on Site C all devices are on the same subnet. There can be a maximum of 1022 devices. Valid network addresses are from 160.40.148.1 to 160.40.151.254. On Site B there are three subnets. Two of the subnets (LAN 1 & LAN 2) can have 1022 devices. Valid network addresses on LAN 1 are 160.40.140.1 to 160.40.143.254. Valid network addresses on LAN 2 are 160.40.152.1 to 160.40.155.254. Also on Site B there is the address space 160.40.156. 0/24, which can have a maximum of 254 devices. Valid network addresses are 160.40.156.1 to 160.40.156.254.

Both the WAN links use the smallest possible subnets to support 2 network addresses by using a mask 255.255.255.252.

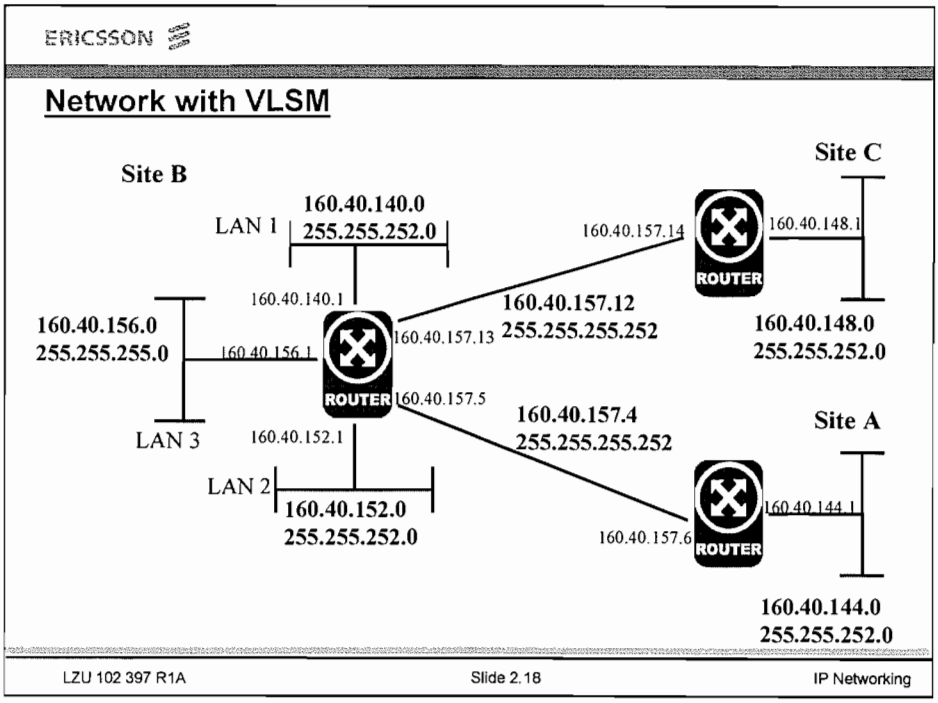


Figure 2-17.

Notes:

AGGREGATION

CIDR supports route aggregation, where a single routing table entry can represent the address space of perhaps thousands of traditional classful routes. This allows a single routing table entry to specify how to route traffic to many individual network addresses. Route aggregation helps control the amount of routing information in the Internet's backbone routers, reduces route flapping (rapid changes in route availability) and eases the local administrative burden of updating external routing information. In the example shown in the diagram assume that an Internet Service Provider (ISP) owns the address block 200.25.0.0/16 . This block represents 65536 (2^{16}) IP addresses (or 256 /24s). From the 200.25.0.0/16 block the ISP wants to allocate the 200.25.16.0/20 address block. This smaller block represents 4,096 (2^{12}) IP addresses (or 16 /24s).

In a classful environment the ISP is forced to cut up the /20 address block into 16 equal size pieces. However, in a classless environment the ISP is free to cut up the address space any way it wants.

It could slice up the address space into 2 equal pieces and assign one portion to company A, then cut the other half into 2 pieces (each 1/4 of the address space) and assign one piece to company B, and finally slice the remaining fourth into 2 pieces (each 1/8 of the address space) and assign one piece each to company C and company D.

Each of the individual companies is free to allocate the address space within its 'intranetwork' as it sees fit.

A prerequisite for aggregating networks' addresses is that they must be consecutive and fall on the correct boundaries.

For example, we cannot aggregate 200.25.24.0/24, 200.25.26.0/24, 200.25.27.0/24 without including the address space 200.25.25.0/24.

CIDR plays an important role in controlling the growth of the Internet's routing tables. The reduction of routing information requires that the Internet be divided into addressing domains. Within a domain, detailed information is available about all the networks that reside in the domain. Outside an addressing domain, only the common network prefix is advertised. This allows a single routing table entry to specify a route to many individual network addresses.

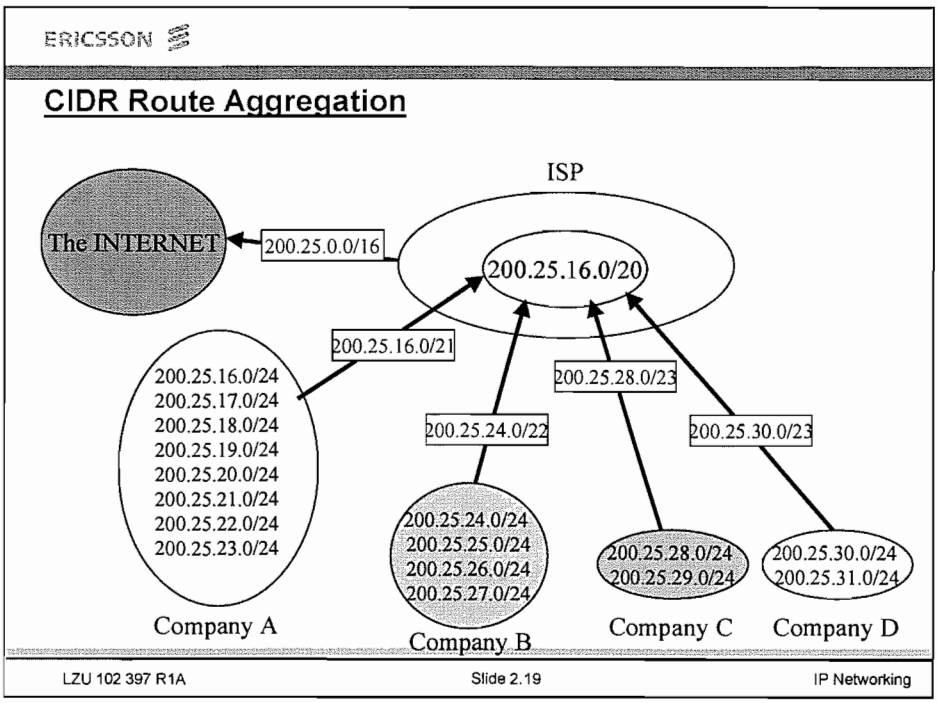


Figure 2-18.

Company A aggregates 8 /24s into single advertisement (200.25.16.0/21).

Company B aggregates 4 /24s into single advertisement (200.25.24.0/22).

Company C aggregates 2 /24s into single advertisement (200.25.28.0/23).

Company D aggregates 2 /24s into single advertisement (200.25.30.0/23).

Finally the ISP is able to inject the 256 /24s in its allocation into the Internet with a single advertisement - 200.25.0.0/16.

IPV6

IP version 6 (IPv6) is a new version of the Internet Protocol, designed as the successor to IP version 4. When IPv4 was defined, only a few computer networks existed. The designers decided to use 32-bit addresses which would allow them to include over a million networks.

The global Internet is, however, growing exponentially, with the size more than doubling annually. At this current rate, all prefixes will soon be assigned and no further growth will be possible using the current address system.

Other reasons for change are new Internet applications. For example, applications that deliver audio and video need to deliver data at regular intervals. To keep such information flowing through the Internet without disruption, IP must avoid changing routes frequently. These new requirements led to the development of IPv6.

The security implemented in IPv6 guarantees that a packet is actually coming from the host indicated in its source address. This is different from IPv4, where the packet could be coming from a host other than that indicated in the source ('spoofing').

Expanded Addressing Capabilities

IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable nodes, and simpler auto-configuration of addresses. The address space provided by IPv6 is large enough to accommodate continued growth of the Internet for many decades. There are enough addresses supported by IPv6 to provide an order of $6 * 10^{23}$ unique addresses per square metre of the surface of the earth.

Header Format Simplification

Some IPv4 header fields have been dropped or made optional. This reduces the processing cost of packet handling and also limits the bandwidth cost of the larger IPv6 header.

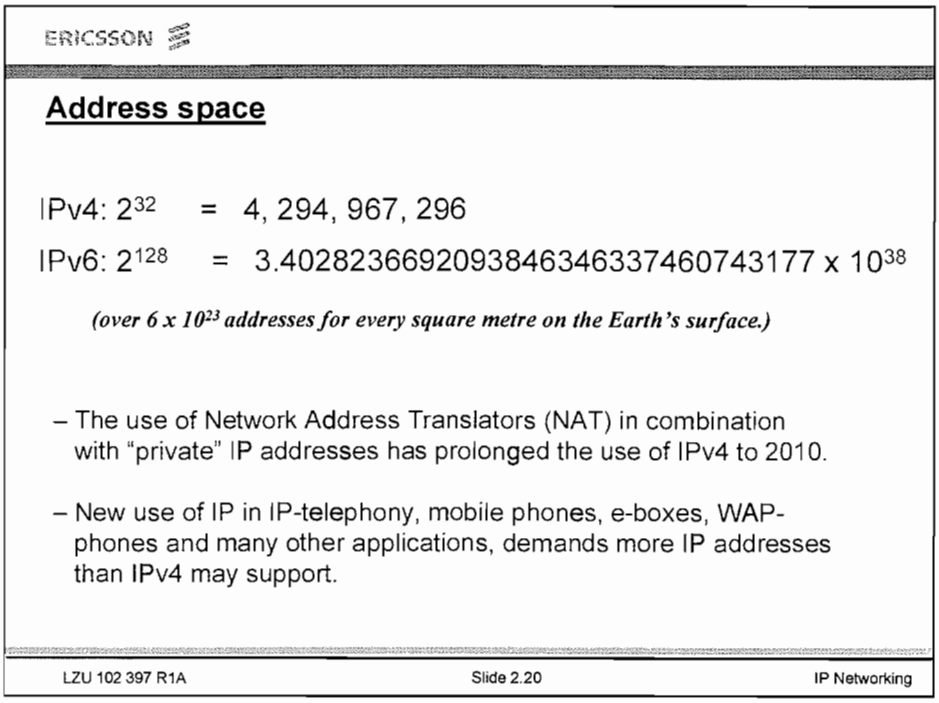


Figure 2-19

Notes:

IPV6 FEATURES:

Improved Support for Extensions and Options

Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future. IPv6 options are placed in separate optional headers that are located between the IPv6 header and the transport layer header. Most of these optional headers are not examined or processed by any router on the packet's path. This simplifies and speeds up router processing of IPv6 packets compared to IPv4 packets.

Flow Labeling Capability

A new capability is added to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling, such as non-default quality of service or "real-time" service for voice or video.

Authentication and Privacy Capabilities

Extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv6.

Address Auto-configuration

This capability provides for dynamic assignment of IPv6 addresses via stateful or stateless address auto-configuration. DHCP is termed a stateful address configuration tool because it maintains static tables that determine which addresses are assigned to new or moved stations. A version of DHCP has been developed for IPv6. IPv6 also supports a stateless address auto-configuration service that does not require a manually configured server. Stateless auto-configuration makes it possible for devices to configure their own addresses with the help of a local IPv6 router. Typically, the device converts it's 48-bit MAC address to an EUI 64-bit format and combines this with a network prefix it learns from a neighbouring router.

Increased Addressing Flexibility

IPv6 includes a new concept of an anycast address, for which a packet is delivered to just one of a set of nodes. The scalability of multicast routing is improved by adding a scope field to multicast addresses.

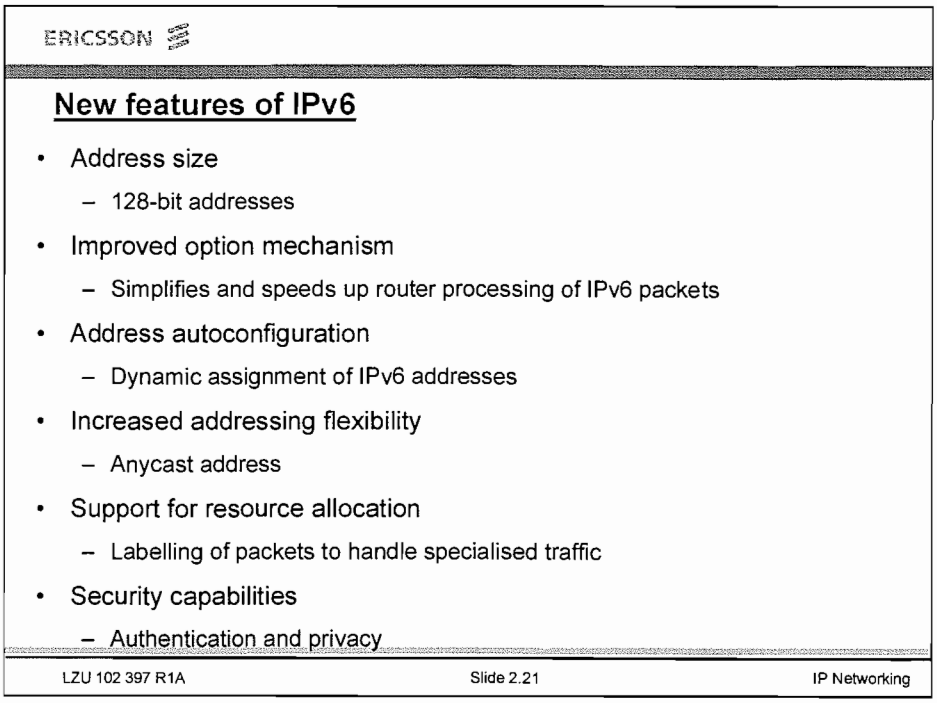


Figure 2-20.

Notes:

IPv6 Packet Format

The IPv6 datagram begins with a base header, which is followed by zero or more extension headers, followed again by data. The only header required is that of the IPv6 header. This is of a fixed size and has a length of 40 octets, compared to 20 octets for the mandatory portion of the IPv4 header. Each header in IPv6 indicates what is following that header. There is no requirement for Internet header length as in IPv4 as each header is a fixed length. Although the IPv6 Header is longer than that of the IPv4 header, it contains fewer fields. Thus routers have less processing to do per header, which should speed up routing.

Version

The 4-bit Internet Protocol version number identifier = 6.

Traffic Class

The 8-bit Traffic Class field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets. The Traffic Class field in the IPv6 header is intended to provide similar functionality to the IPv4 Type of Service or Precedence bits for various forms of "differentiated service".

Flow Label

The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. This aspect of IPv6 may be subject to change, as the requirements for flow support in the Internet become clearer. Hosts or routers that do not support the functions of the Flow Label field are required to set the field to zero when originating a packet, pass the field on unchanged when forwarding a packet, and ignore the field when receiving a packet.

Payload Length

The 16-bit unsigned integer indicates the length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets. (Note that any extension headers present are considered part of the payload, and are included in the length count.)

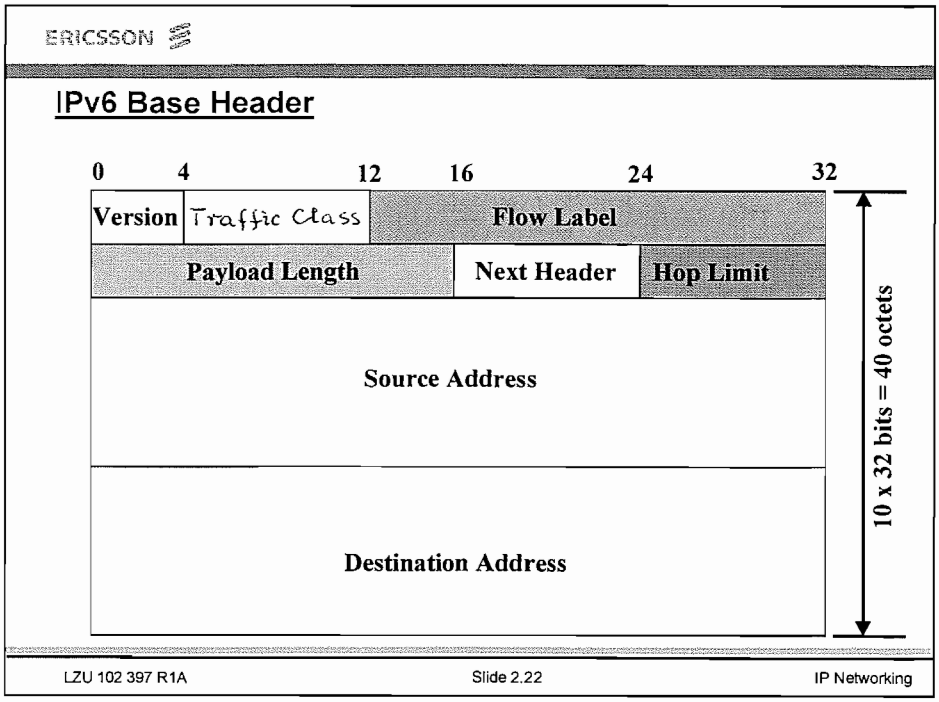


Figure 2-21.

Notes:

IPv6 Packet Format (cont)

Next Header

This 8-bit selector identifies the type of header immediately following the IPv6 header or subsequent header in the packet. It uses the same values as the IPv4 Protocol field.

See http://www.iana.org/assignments/protocol-numbers for protocol values.

Hop Limit

This is an 8-bit integer. It is decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero. It may be used to count the number of routers visited.

Source Address

The source 128-bit address of the originator of the packet.

Destination Address

The 128-bit address of the intended recipient of the packet. The address may not be the ultimate recipient if a Routing Header is present. At the originating node, that ultimate address will be in the last element of the Routing Header and at the recipient, that ultimate address will be in the Destination address field of the IPv6 header.

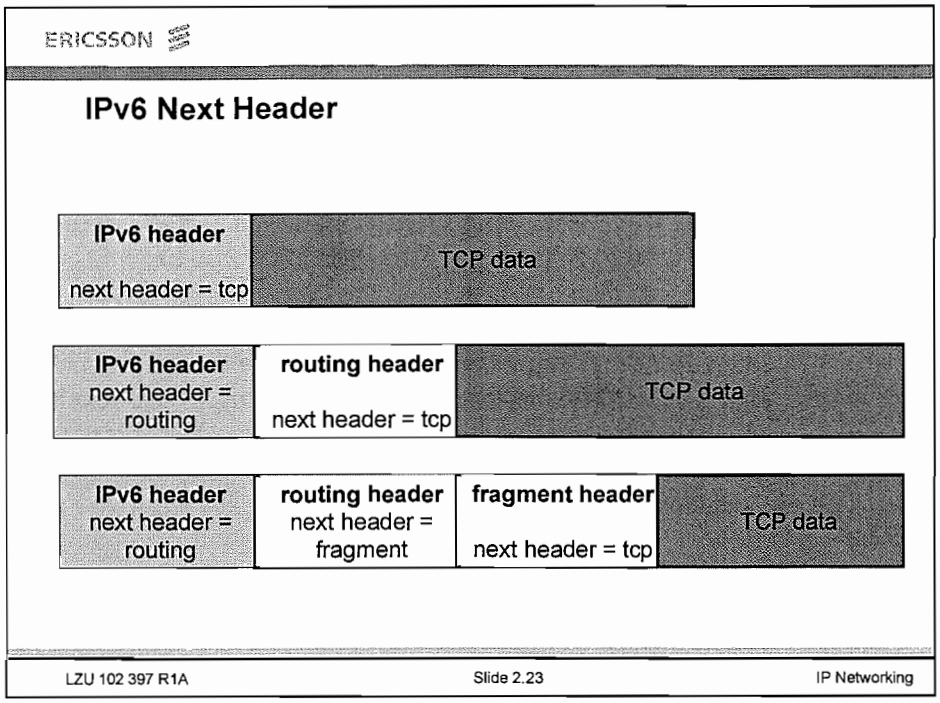


Figure 2-22.

Notes:

IPv6 Extension Headers

There are seven kinds of extension header in IPv6. These headers are supplied to provide extra information, but are encoded in an efficient way. Processing speed is increased because each header is a fixed length. Each one of the seven extension headers is optional and if more than one is present, they must appear directly after the fixed header and in the preferred order. The IPv6 header and extension header contain a Next Header field. This field identifies the type of the header immediately following. If the next header is an extension, then this field contains the type identifier of that header. Otherwise, the Next Header field contains the protocol identifier of the upper-layer protocol using IPv6.

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 the case of multicast) identified in the Destination Address field of the IPv6 header. There, normal de-multiplexing 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.

ERICSSON

Assigned Internet Protocol Numbers

Decimal	Keyword	Protocol	
0		HOPOPT	IPv6 Hop-by-Hop Option [RFC1883]
1		ICMP	Internet Control Message [RFC792]
6		TCP	Transmission Control [RFC793]
17		UDP	User Datagram [RFC768]
41		IPv6	Ipv6
43		IPv6-Route	Routing Header for IPv6
44		IPv6-Frag	Fragment Header for IPv6
50		ESP	Security Payload for IPv6 [RFC2406]
51		AH	Authentication Header for IPv6 [RFC2402]
58		IPv6-ICMP	ICMP for IPv6 [RFC1883]
60		IPv6-Opts	Destination Options for IPv6 [RFC1883]

LZU 102 397 R1A Slide 2.24 IP Networking

Figure 2-23.

Notes:

IPv6 Extension Header Descriptions

Hop-by-hop Options Header

Defines special options that require hop-by-hop processing

Destination Option Header 1

Contains optional information to be examined by the first destination listed in the IPv6 address field. This header can also be read by a subsequent destination listed in the source routing header address fields.

Routing Header

Allows a source node to specify a list of IP addresses that dictate what path a packet will traverse.

Fragment Header

Contains fragmentation and reassembly information.

Authentication Header

Provides packet integrity and authentication.

Encapsulated Security Payload Header

Provides encryption.

Destination Options Header 2

Contains additional information to be examined only by the final destination node.

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.

ERICSSON

IPv6 Extension Header

Extension header	Description
Hop-by-hop options	Miscellaneous information for routers
Destination options -1	Information for 1st destination
Routing Header	Full or partial route to follow
Fragmentation	Management of datagram fragments
Authentication	Verification of the sender's identity
Encrypted security payload	Information about the encrypted contents
Destination options -2	Additional information for the final destination only

LZU 102 397 R1A Slide 2.25 IP Networking

Figure 2-24.

Notes:

IPv6 Routing Header

The routing header allows a source node to specify a list of IP addresses that dictate what path a packet will traverse.

RFC 2460 defines a version of this routing header called 'Type 0', which gives a sending node a great deal of control over each packet's route. Type zero routing headers contain a 24-bit field that indicates how intermediate nodes may forward a packet to the next address in the routing header.

Each bit in the 24-bit field indicates whether the next corresponding destination address must be a neighbour of the preceding address (1 indicates 'strict, must be a neighbour', and 0 indicates 'loose, need not be a neighbour').

When routing headers are used for 'strict' forwarding, a packet visits only routers listed in the routing header.

For example, if routers 2 and 3 are listed as strict but are not adjacent to each other (that is, in order to get from 2 to 3 a packet must pass through some other router), packets are dropped at 2. This is a valuable feature when security and traffic control require that packets take a rigidly defined path.

In 'loose' forwarding, a packet can visit unlisted routers. When Type 0 routing headers are used, the initial IPv6 header contains the destination address of the first router in the source route, not the final destination address.

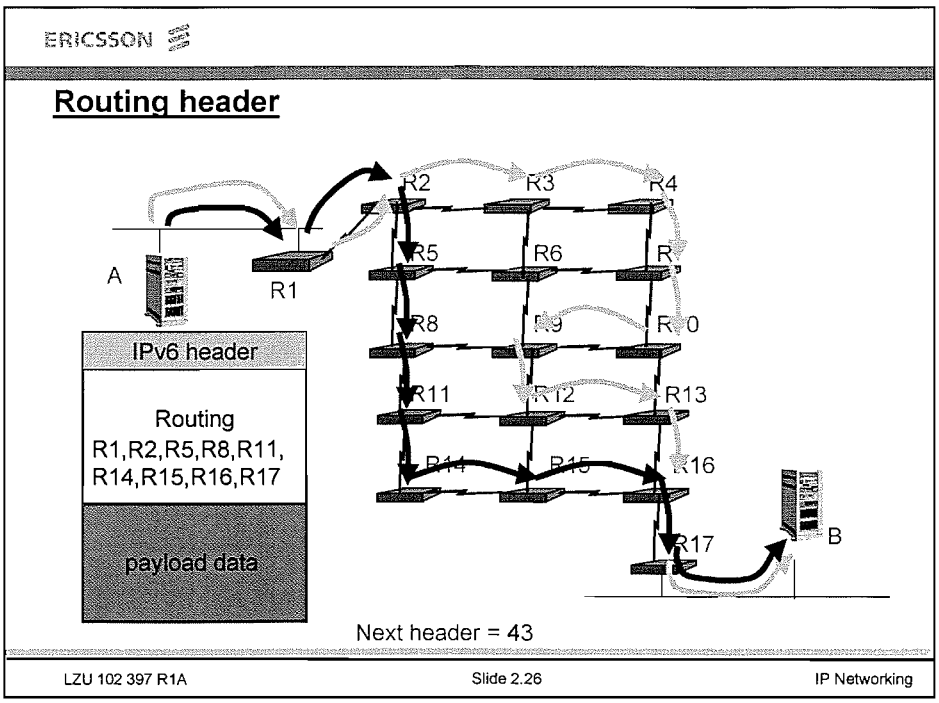


Figure 2-25.

Notes:

Fragment Header

In IPv6, fragmentation may only be performed by source nodes and not by routers along a packet's delivery path. End nodes performing fragmentation can determine the smallest MTU of a path using the MTU path discovery process. IPv6 requires that every link in the Internet have an MTU of 1280 octets or greater.

The source node sends out a packet with an MTU as large as the local interface can handle. If this MTU is too large for some link along the path, an ICMP 'datagram too big' message is sent back to the source. This message contains a 'datagram too big' indicator and the MTU of the affected link.

The source can then adjust the packet size downward and retransmit another packet. This process is repeated until a packet gets all the way to the destination node. The discovered MTU is then used for fragmentation purposes.

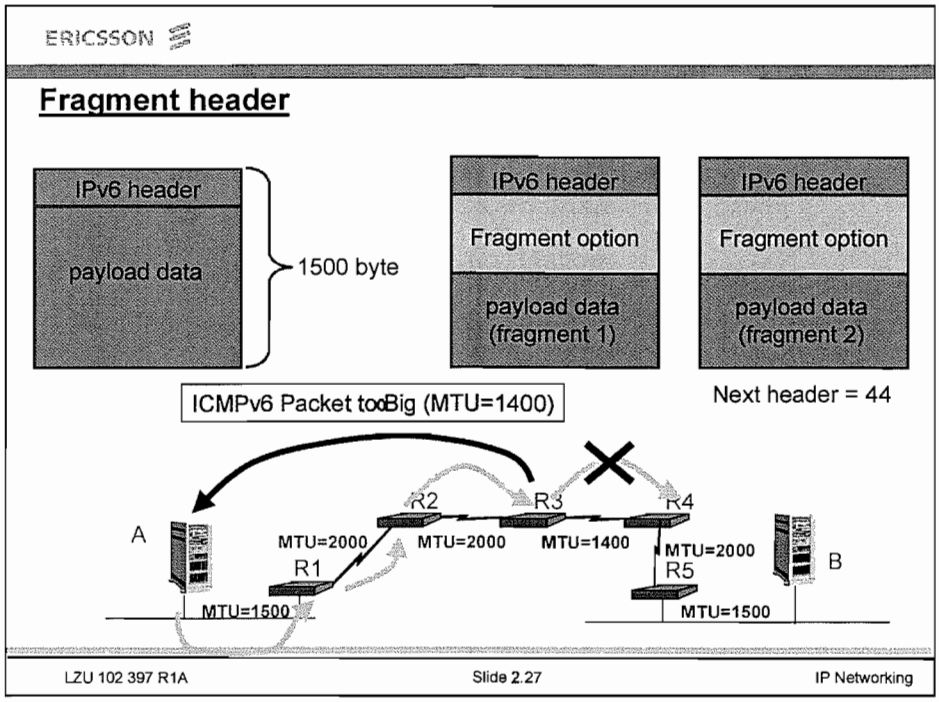


Figure 2-26.

Notes:

Hop-by-Hop Options Header

The hop-by-hop option header carries optional information that, if present, must be examined by every router along the path. This header consists of the following:

Next Header (8 bits): identifies the type of header immediately following this header.

Header Extension Length (8 bits): the length of the current header in 64-bit units, not including the first 64 bits.

Options: a variable length field consisting of one or more option definitions. Each definition is in the form of three subfields:

Option type (8 bits), which identifies the option

Length (8 bits), which specifies the length of the option data field in octets

Option data, which is a variable-length specification of the option.

Destination Options Headers

There are two variations of this header, each with a different position in the packet. The first variation is for carrying information to the first destination listed in the IPv6 address field.

This header can also be read by a subsequent destination listed in the source routing header address fields.

The second variation is used for optional information that is only to be read by the final destination.

For efficiency, the first variation is typically located towards the front of the header chain, directly after the hop-by-hop header (if any). The second variation is relegated to a position at the end of the extension header chain, which is typically the last IPv6 optional header before transport and payload.

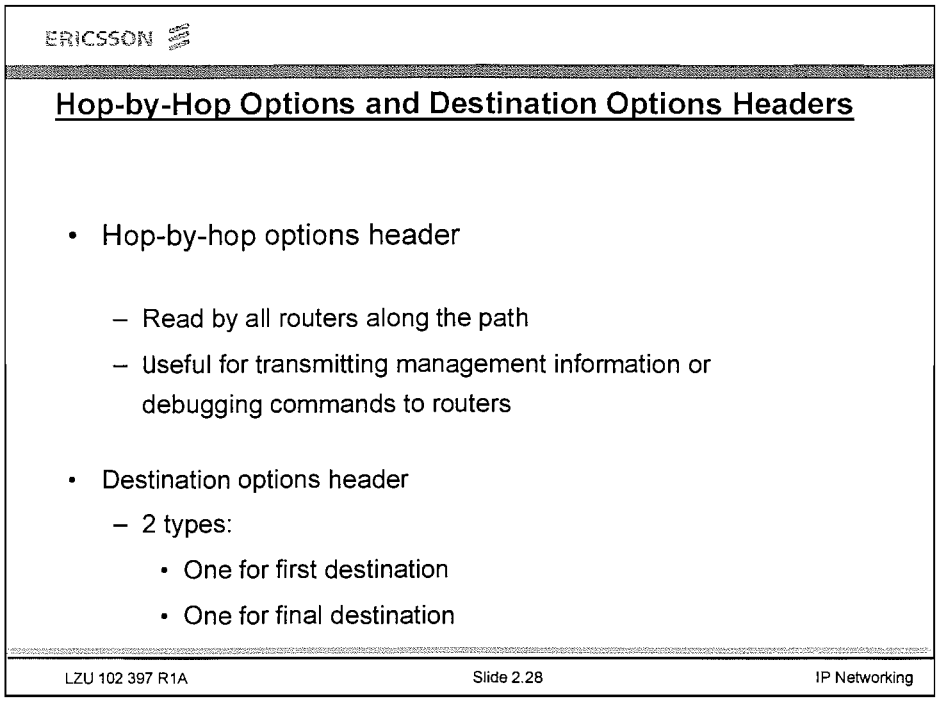


Figure 2-27.

Notes:

Authentication Header

The IPv6 authentication extension header gives network applications a guarantee that the packet did in fact come from an authentic source as indicated in its source address. This authentication is particularly important to safeguard against intruders who configure a host to generate packets with forged source addresses.

This type of source address masquerading can spoof a server, with the result that access may be gained to valuable data, passwords or network control utilities. With IPv6 authentication headers, hosts establish a standards-based security association that is based on the exchange of algorithm-independent security keys (for example MD5).

In a client/server session, for instance, both the client and the server must know the key. Before each packet is sent, IPv6 authentication creates a checksum based on the key combined with the entire contents of the packet. This checksum is then re-run on the receiving side and compared.

This approach provides authentication of the sender and guarantees that an intervening party has not modified data within the packet. Authentication can take place between clients and servers or clients and clients on the corporate backbone. It can also be deployed between remote stations and corporate dialin servers to ensure that the perimeter of the corporate security is not breached.

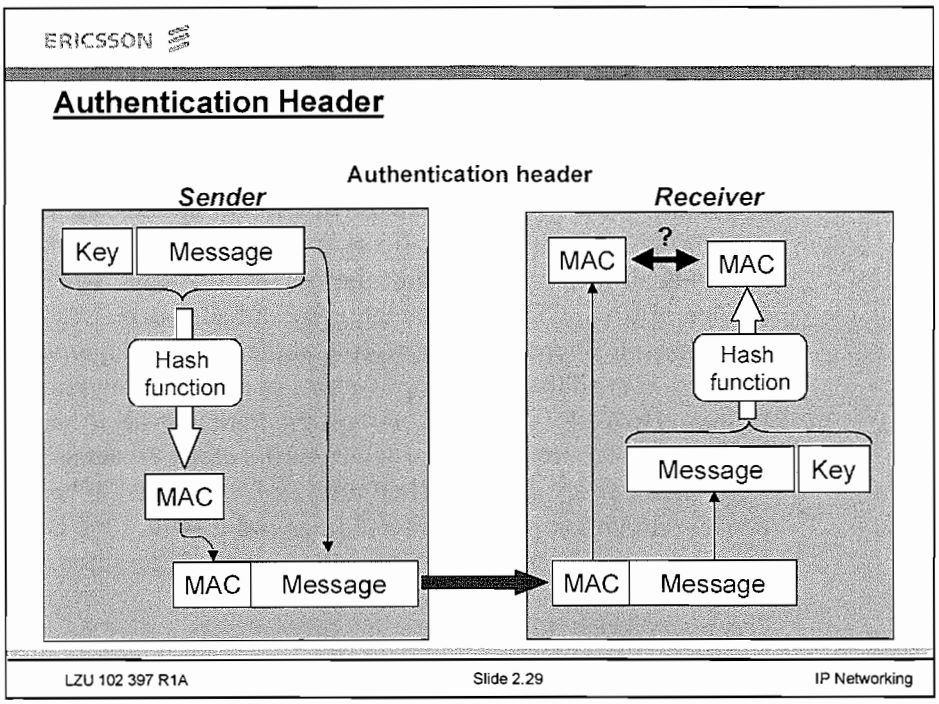


Figure 2-28.

Notes:

Encryption Security Payload Header

Authentication headers eliminate a number of host spoofing and packet modification hacks. They do not, however, prevent nondisruptive reading, such as sniffing (reading network traffic), as data traverses the Internet and corporate backbone networks. This area is dealt with by the Encapsulating Security Payload (ESP) service of IPv6. ESP provides integrity and confidentiality in IPv6 datagrams. ESP provides encryption at the network layer, making encryption available to all applications in a highly standardised fashion. IPv6 ESP is used to encrypt the transport-layer header and payload (for example, TCP or UDP) or the entire IP datagram. Both these methods are accomplished with an ESP extension header that carries encryption parameters and keys end-to-end. IPv6 has two modes to provide confidentiality: transport mode and tunnel mode.

Transport Mode ESP

In this mode only the payload is encrypted. The IP header and IP options are unencrypted and are used for routing the packet. The receiver decrypts the ESP and continues to use the unencrypted header as an IP header if necessary.

Tunnel Mode ESP

In this mode, the original IP datagram and header are encrypted. This entire ESP frame is placed within a new datagram with an unencrypted IP header. All additional unencrypted information, such as routing header, is placed between the IP header and the encapsulated security payload. The receiver strips off the cleartext IP header and decrypts the ESP.

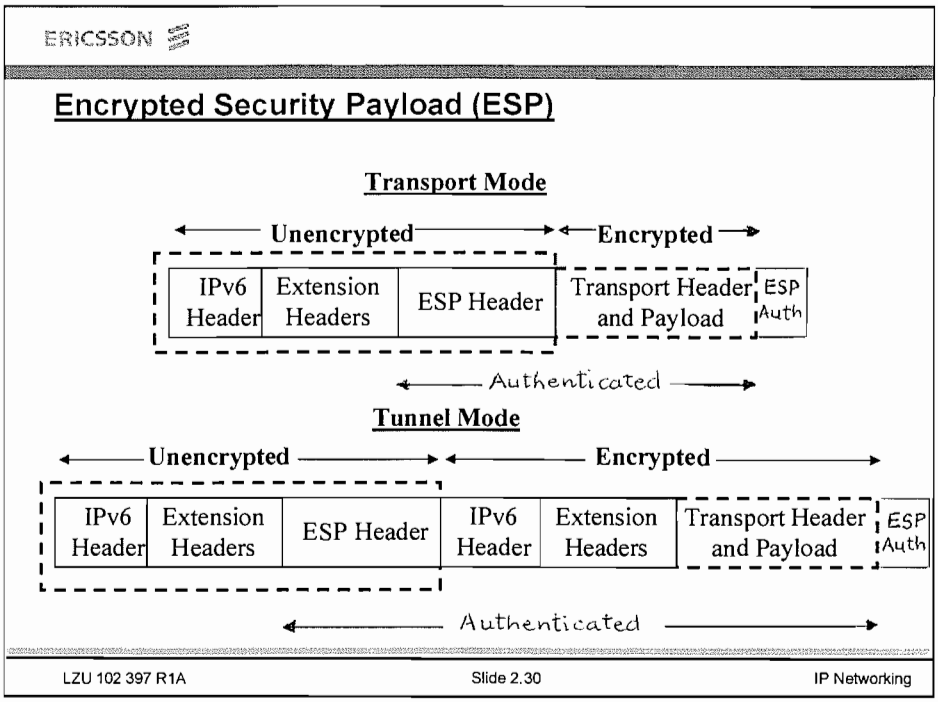


Figure 2-29.

Notes:

IPV6 ADDRESSING

Like IPv4, IPv6 assigns a unique address for each connection between a computer and a physical network. Thus, if a computer connects to three physical networks, the computer is assigned three addresses. IPv6 also separates each such address into a prefix that identifies the network and a suffix that identifies a particular computer on the network. Although IPv6 adopts the same approach for assigning computer addresses, IPv6 addressing differs from IPv4 in the following ways:

With IPv6, all address details are completely different.

IPv6 defines a set of special addresses that differ considerably from IPv4 special addresses. In particular, IPv6 does not include a broadcast address. Instead, it uses a form of multicasting.

There are three types of IPv6 addresses: unicast, multicast, and anycast.

Unicast

The address corresponds to a single computer. A datagram sent to the address is routed along the shortest path to the computer.

Multicast

The address corresponds to a set of computers, possibly at many locations. Membership in the set can change at any time. When a datagram is sent to the address, IPv6 delivers one copy of the datagram to each member of the set.

Anycast

The address corresponds to a set of computers that share a common address prefix. A datagram sent to the address is routed along the shortest path and delivered to one of the computers, typically the 'nearest' one according to current routing protocol metrics.

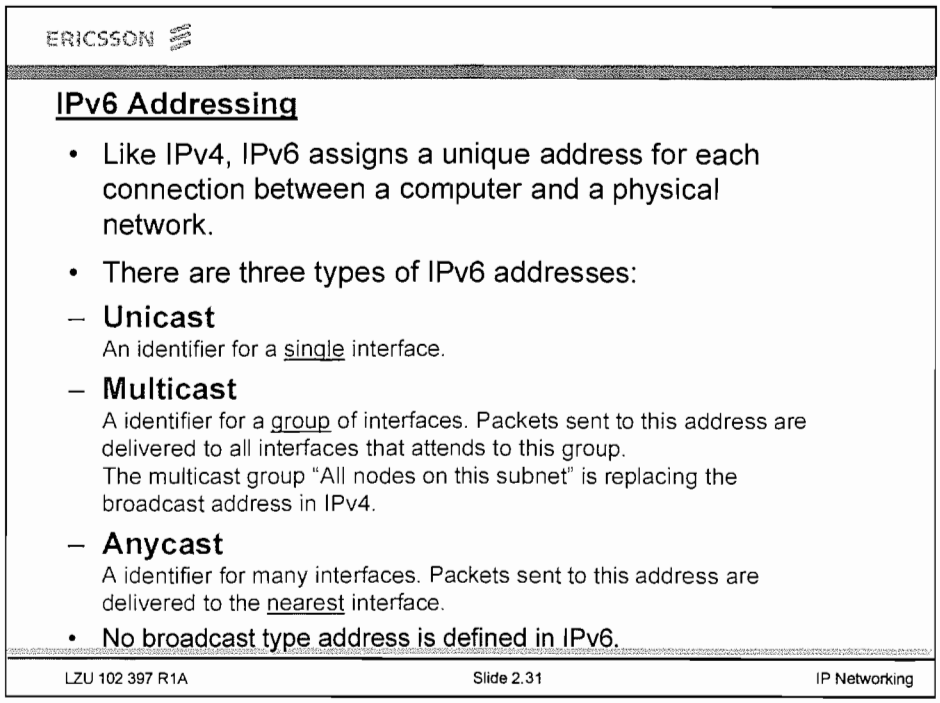


Figure 2-30.

Notes:

Unicast Addresses

There are several types of unicast addresses in IPv6, in particular global unicast, site-local unicast, and link-local unicast. There are also some special-purpose subtypes of global unicast, such as IPv6 addresses with embedded IPv4 addresses or encoded NSAP addresses.

Link Local

The Link-Local is for use on a single link or network. Link-Local addresses are designed to be used for addressing on a single link for purposes such as auto-address configuration, neighbour discovery, or when no routers are present. You can have the same link local address on different networks. Routers must not forward any packets with link-local source or destination addresses to other links.

Site Local

Site-Local addresses are designed to be used for addressing inside of a site without the need for a global prefix. It could typically be an Intranet for an organisation. Routers must not forward any packets with site-local source or destination addresses outside of the site. You can choose addresses that are only used inside the organisation like the private addresses in IPv4: 192.168.. 172... 10. For security: if everybody uses site local addresses inside an organisation nothing will be sent out of the local network.

Global Unicast

Addresses are used for global communication. These addresses are aggregatable to ensure that the routing table in the core routers in the Internet won't grow unrestrained. There is a lot of focus of how to allocate these addresses to keep an optimal aggregation. Multi-homing where you have more than one ISP break the possibility for efficient aggregation.

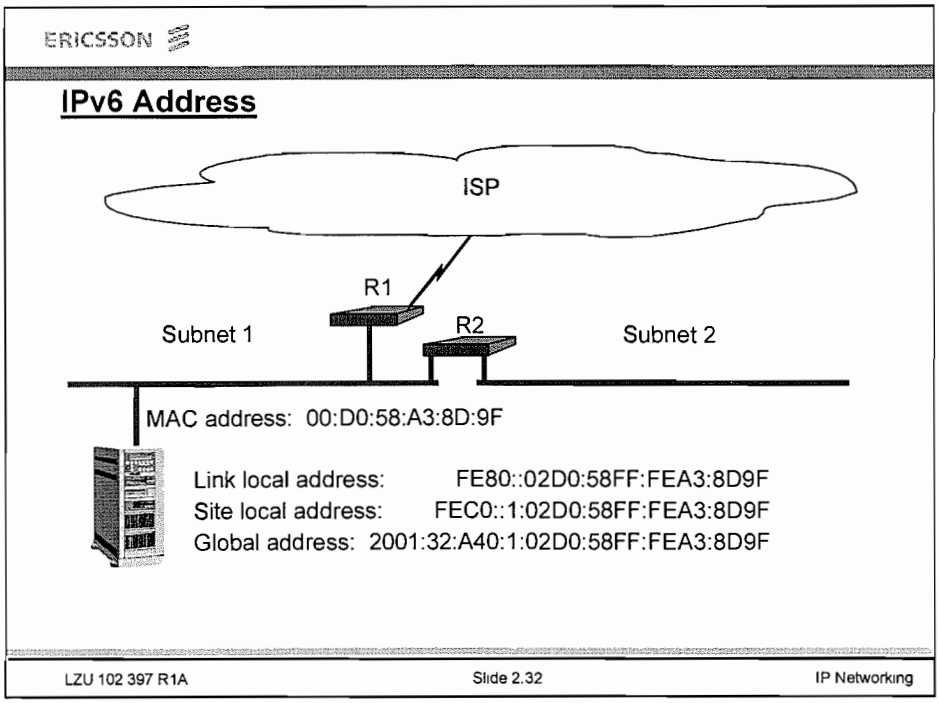


Figure 2-31.

Notes:

IPv6 Colon Hexadecimal Notation

Writing 128-bit numbers can be confusing. For example, consider a 128-bit number written in dotted decimal notation:

105.220.136.100.255.255.255.255.0.0.18.128.140.10.255.255

To help reduce the number of characters in an address, the designers of IPv6 propose using a more compact syntactic form known as hexadecimal notation. With hexadecimal notation, each group of 16 bits is written in hexadecimal with a colon separating groups. For example, when the above number is written in colon hex, it becomes:

69DC:8864:FFFF:FFFF:0:1280:8C0A:FFFF

As illustrated in the example, colon hex notation requires fewer characters to express an address. An additional optimisation known as zero compression further reduces the size. Zero compression replaces sequences of zeroes with two colons. For example, the address:

FF0C:0:0:0:0:0:0:B1 can be written as: FF0C::B1

Note that this zero compression may be used only one instance in an address as it replaces an unknown number of zeros. The "::" can also be used to compress leading or trailing zeros in an address

The address 0:0:0:0:0:0:0:0 (::) is called the unspecified address. It must never be assigned to any node. It indicates the absence of an address. This address is sent as the Source Address field of any IPv6 datagrams sent by an initializing host before it has learned its own address.

The Unicast address 0:0:0:0:0:0:0:1 (::1) is called the loopback address. It may be used by a node to send an IPv6 datagram to itself. It must never be assigned to any interface or be sent outside of a single node.

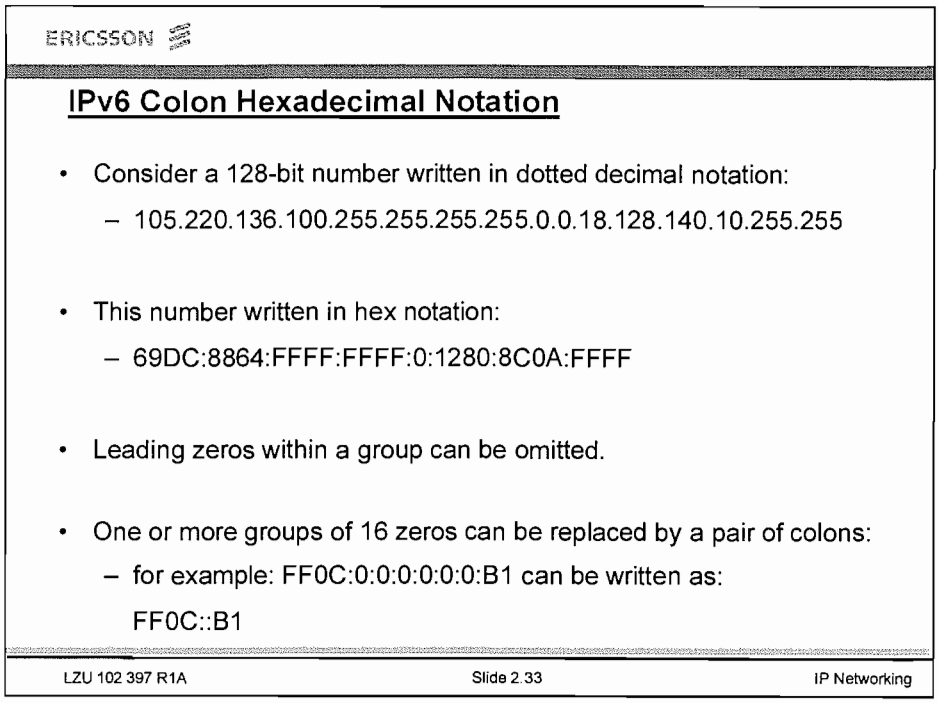


Figure 2-32.

Notes:

IPV6 ADDRESS FORMAT

The format for IPv6 global aggregatable unicast addresses is as follows:

FP

This is the Format Prefix used to identify aggregatable global unicast addresses:

- Global Unicast Addresses **001**x (02x or 03x)
- Link-Local Unicast Addresses **1111 1110 10** (FE80).
- Site-Local Unicast Addresses **1111 1110 11** (FEC0).
- Multicast Addresses **1111 1111** (FF).

TLA

Top-Level Aggregation Identifier is 13 bits. This allows for 8,192 (2^{13}) Top-Level Aggregation Identifiers. TLA contains routing information for top level routers in the Internet and the number is set to ensure that the top level routes is kept within this number.

RES

This 8-bit field is reserved for significant growth of either the TLA or NLA fields.

NLA

The Next Level Aggregator is used by organisations that control top level aggregations ID to create an addressing hierarchy and to identify sites.

SLA

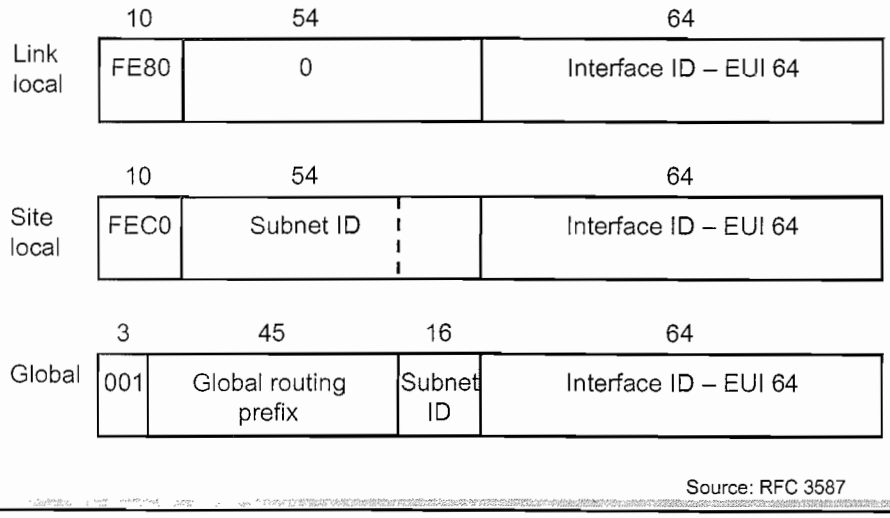
The Site-Level Aggregation is a 16-bit field that allows organisations to create a local addressing hierarchy. SLA supports 65,535 (2^{16}) individual subnets per site.

Interface Identifier:

Is a 64-bit field based on the IEEE EUI-64 address, which identifies interfaces on a link.

ERICSSON

Address format



Source: RFC 3587

LZU 102 397 R1A Slide 2.40 IP Networking

Notes:

Interface Identifier

The use of IEEE 802 MAC addresses as an Interface ID is expected to be very common in environments where nodes have an IEEE 802 MAC address. Using the globally unique MAC address makes possible a simple form of auto-configuration of addresses. In other environments, where MAC addresses are not available, other types of link layer addresses will be used as Interface ID.

Ethernet hardware addresses are 48 bits, expressed as 12 hexadecimal digits. These 12 hex digits consist of the first/left 6 digits – Company ID, allocated by IEEE - and the last/right 6 digits which specify the interface serial number for that interface vendor. The vendor must control these numbers so that no network cards have got the same number.

Ethernet addresses may be written un-hyphenated (e.g. 123456789ABC), or with one hyphen (e.g. 123456-789ABC), but should be written hyphenated by octets (e.g. 12-34-56-78-9A-BC). These addresses are physical station addresses, not multicast or broadcast, so the second hex digit (reading from the left) will be even, not odd.

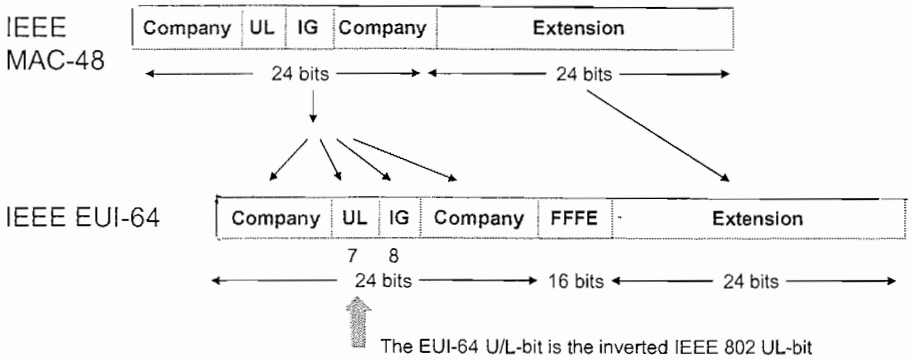
IPv6 host addresses are based on the IEEE EUI-64 format (64 bits) for interface identifiers that are built on basics of the 48 bits MAC address (IEEE 802). Converting from the 48-bit MAC address to the EUI-64 address, the fixed value FFFE (16 bits) are inserted between the Company ID and Extension ID.

EUI-64 based Interface identifiers may have global scope when a global identifier is available (e.g., IEEE 48bit MAC) or may have local scope where a global identifier is not available (e.g., serial links, tunnel end-points, etc.). The motivation for inverting the "u" bit when forming an interface identifier is to make it easy for system administrators to hand configure non-global identifiers when hardware tokens are not available. This is expected to be case for serial links, tunnel end- points, etc. The alternative would have been for these to be of the form 0200:0:0:1, 0200:0:0:2, etc., instead of the much simpler 1, 2, etc.

ERICSSON

Address identifier

- Conversion from IEEE MAC-48 to IEEE EUI-64



LZU 102 397 R1A Slide 2.41 IP Networking

Notes:

IPv6 Addresses with Embedded IPv4 Addresses

The IPv6 transition mechanisms include a technique for hosts and routers to dynamically tunnel IPv6 packets over IPv4 routing infrastructure. IPv6 nodes that utilize this technique are assigned special IPv6 unicast addresses that carry an IPv4 address in the low-order 32 bits. This type of address is termed as "IPv4-compatible IPv6 address".

A second type of IPv6 address, which holds an embedded IPv4 address, is also defined. This address is used to represent the addresses of IPv4-only nodes (nodes, which do not have a dual protocol, stack and do not support IPv6) as IPv6 addresses. This type of address is termed 'IPv4-mapped IPv6 address'.

A third type of IPv6 address, which holds an embedded IPv4 address, is utilized by some transition mechanisms. The 'IPv4-translated' address is used by an IPv6-enabled node when addressing an IPv4 node through an IPv6 - IPv4 protocol translator.

6-to-4 Addresses

This type of address allows isolated IPv6 sites to communicate through an IPv4 backbone. The sites must have at least one globally unique IPv4 address.

The IPv6 packets from a 6-to-4 site are encapsulated in IPv4 packets when they leave the site via its external IPv4 connection.

All 6-to-4 addresses have a 2002::/16 address prefix. (RFC 3056)

The IPv4 header contains the destination and source IPv4 addresses, protocol type 41, and the IPv6 header and payload are carried as IPv4 payload. IPv6 sites connected using this method do not require IPv4-compatible IPv6 addresses. The 6-to-4 mechanism should be implemented in border routers with an address selection algorithm.

A site which has an IPv4 address and is able to send IPv4 packets with protocol type 41 creates a DNS entry with the IPv6 prefix {0x2002, IPv4 address}.

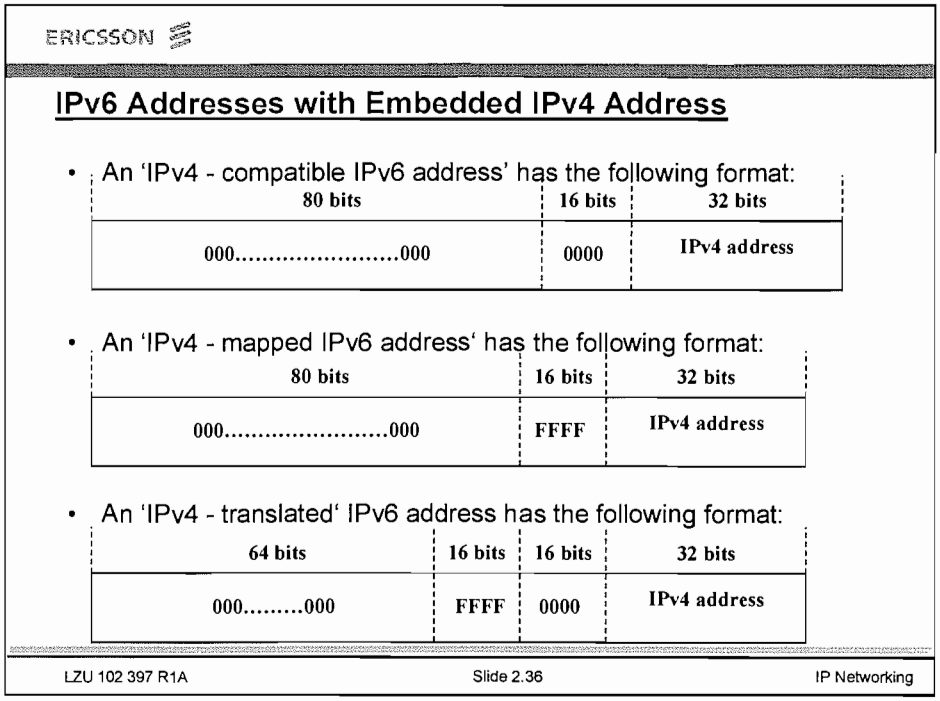


Figure 2-35.

Notes:

Transition to IPv6

Once the IP addresses of the end points have been determined, appropriate routing mechanisms are necessary to send IP packets back and forth. If both the sender and the recipient have standard IPv6 addresses and direct connections to an IPv6 backbone, routing is straightforward. If they can reach each other only over an IPv4 network, IPv4 encapsulation is necessary while traversing the IPv4 part of the network. If each end supports a different version of IP, then a protocol translator or gateway is needed between them.

IPv6 hosts and routers will need to retain backward compatibility with IPv4 devices for an extended time period (possibly years or even indefinitely) and will probably have the option of retaining their IPv4 addressing. To accomplish these goals, IPv6 transition relies on several special functions that have been built into the IPv6 standardisation work, including dual-stack hosts and routers, transition mechanisms that temporarily assign an IPv4 address to an IPv6 host, and tunneling IPv6 via IPv4.

Dual Stack.

Once a few nodes have been converted to IPv6, there is the strong possibility that these nodes will require continued interaction with existing IPv4 nodes. This is accomplished using the dual-stack IPv4/IPv6 approach. When running a dual IPv4/IPv6 stack, a host can access both IPv4 and IPv6 resources. Routers running both protocols can forward traffic for both IPv4 and IPv6 end nodes. Dual stack machines can use totally independent IPv4 and IPv6 addresses, or they can be configured with an IPv6 address that is IPv4 compatible.

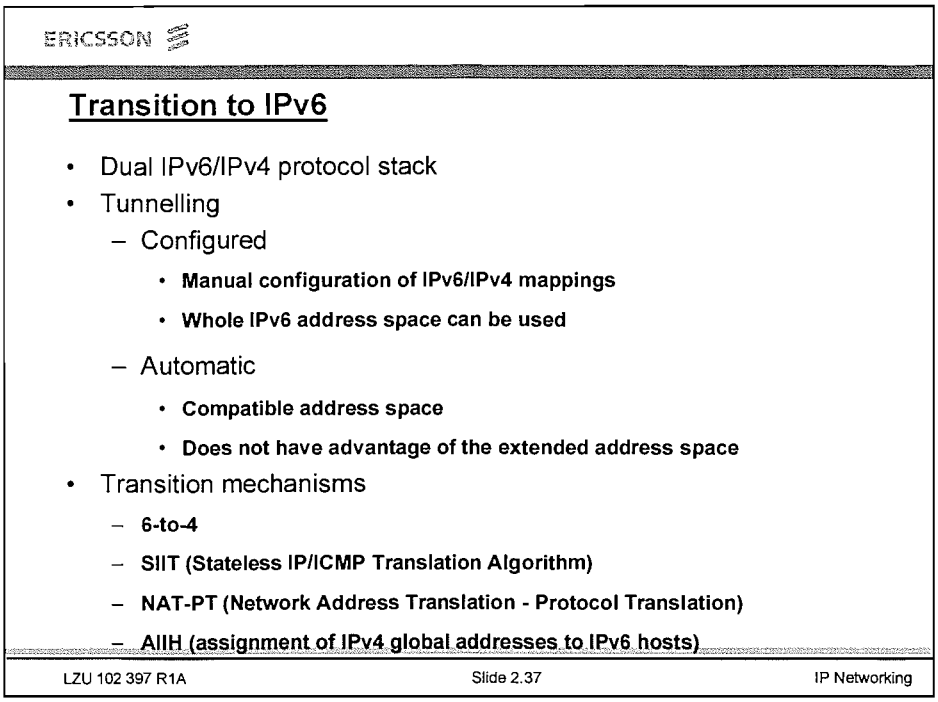


Figure 2-36.

Notes:

IPv6 over IPv4 Tunnelling

In most deployment scenarios, the IPv6 routing infrastructure is built up over time. While the IPv6 infrastructure is being deployed, the existing IPv4 routing infrastructure can remain functional, and can be used to carry IPv6 traffic.

Tunnelling provides a way to utilize an existing IPv4 routing infrastructure to carry IPv6 traffic. To be able to carry an IPv6 packet over an IPv4 backbone, an IPv4 header is added to the packet. The value of the protocol field in the appended IPv4 header is set to 41, in order to indicate that the packet contains an encapsulated packet.

Tunnelling techniques are usually classified according to the mechanism by which the encapsulating node determines the address of the node at the end of the tunnel. If the tunnel endpoint is the node to which the IPv6 packet is addressed, the tunnel endpoint can be determined from the destination IPv6 address of that packet. If that address is an IPv4-compatible IPv6 address, then the low-order 32 bits hold the IPv4 address of the destination node, and that can be used as the tunnel endpoint address. This technique avoids the need to explicitly configure the tunnel endpoint address. Deriving the tunnel endpoint address from the embedded IPv4 address of the packet's IPv6 address is termed 'automatic tunnelling'. This technique can only be applied if the IPv6 hosts use IPv4-compatible IPv6 addresses.

If the endpoint of the tunnel is an intermediary router, which must decapsulate the IPv6 packet and forward it to the final destination, the endpoint of the tunnel is different from the destination of the packet being tunnelled. So the addresses in the IPv6 packet being tunnelled do not provide the IPv4 address of the tunnel endpoint, even if IPv4-compatible IPv6 addresses are in use. In this case, the tunnel endpoint address must be determined from configuration information on the node performing the tunnelling. This technique is termed 'configured tunnelling' and can be applied to any type of IPv6 address.

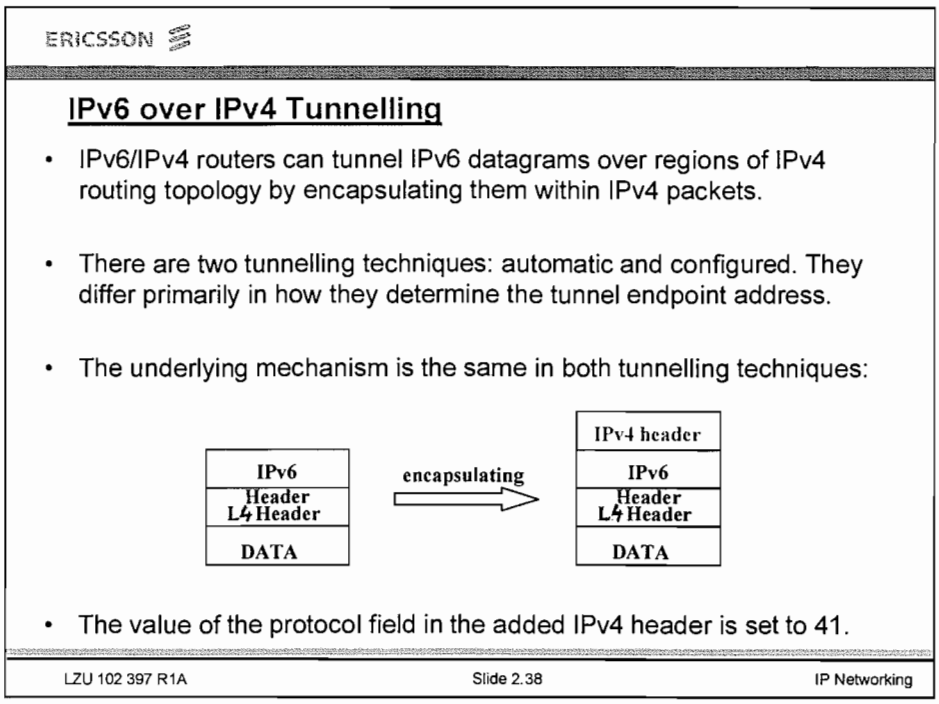


Figure 2-37.

Notes:

ICMP ERROR MESSAGES AND HOP COUNT FIELD

The fragmentation inside the tunnel can be reduced to a minimum by having the encapsulating node track the IPv4 path MTU across the tunnel. This is done by using the IPv4 Path MTU Discovery Protocol (RFC 1191) and by recording the resulting path MTU.

The IPv6 layer in the encapsulating node can then view a tunnel as a link layer with an MTU equal to the IPv4 path MTU, minus the size of the encapsulating IPv4 header (usually 20 bytes).

In response to encapsulated packets it has sent into the tunnel, the encapsulating node may receive IPv4 ICMP error messages from IPv4 routers inside the tunnel. These packets are addressed to the encapsulating node because it is the IPv4 source of the encapsulated packet.

The handling of other types of ICMP error messages depends on how much information is included in the 'packet in error' field, which holds the encapsulated packet that caused the error. If the offending packet includes enough data, the encapsulating node may extract the encapsulated IPv6 packet and use it to generating an IPv6 ICMP message directed back to the originating IPv6 node.

IPv6-over-IPv4 tunnels are modelled as 'single-hop'. That is, the IPv6 hop limit is decremented by 1 when an IPv6 packet traverses the tunnel. The single-hop model serves to hide the existence of a tunnel. The tunnel is opaque to users of the network, and is not detectable by network diagnostic tools such as traceroute. The TTL of the encapsulating IPv4 header is selected in an implementation-dependent manner and decremented as in pure IPv4 networks.

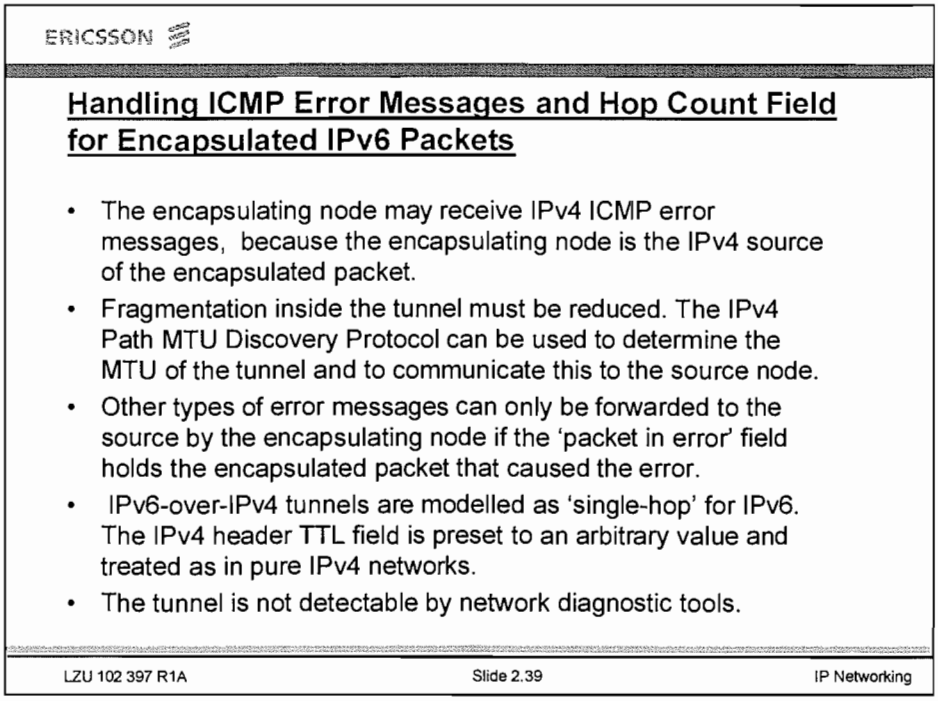


Figure 2-38.

Notes:

Automatic Encapsulation: the 6-to-4 Mechanism

In the figure opposite, an IPv6-only node B, having a 64-bit interface identifier '260:97FF:FEA8:E5A8' and a 16-bit site-level aggregator 'F00', is connected to the IPv4 Internet via a dual stack 6-to-4 router M.

M has a publicly routable IPv4 address '192.1.2.3' or 'C001:0203' in hexadecimal notation. The 6-to-4 address of B (a valid IPv6 address, in fact) is '2002:C001:0203:F00:260:97FF:FEA8:E5A8'.

Similarly, the IPv6-only node A is connected to the IPv4 Internet via the dual stack 6-to-4 router N, which has a publicly routable IPv4 address '9.254.251.252' or '09FE:FBFC'.

The 6-to-4 address of A is '2002:09FE:FBFC:A00:140:98AA:ABC8:A3B7'.

When a sending host or router (such as B or M) sees a packet with the destination address of A, it first extracts the embedded IPv4 address (in this case '09FE:FBFC'), and encapsulates the IPv6 packet in an IPv4 packet destined for this embedded address. When the 6-to-4 router N receives this packet, it decapsulates it, and forwards it to A using native IPv6 routing within the IPv6 stub network. This mechanism is an alternative solution to automatic encapsulation. Nodes wishing to communicate using 6-to-4 addresses must satisfy the following restrictions:

Each 6-to-4 router must have at least one publicly routable IPv4 address. The so-called '6-to-4' automatic encapsulation mechanism reduces this requirement to just a single publicly routable IPv4 address per site.

Each node behind a 6-to-4 router with the 'a.b.c.d' IPv4 address must have an IPv6 address of 2002:ab:cd:SLA:I_faceID, with SLA and Interface_ID of the site.

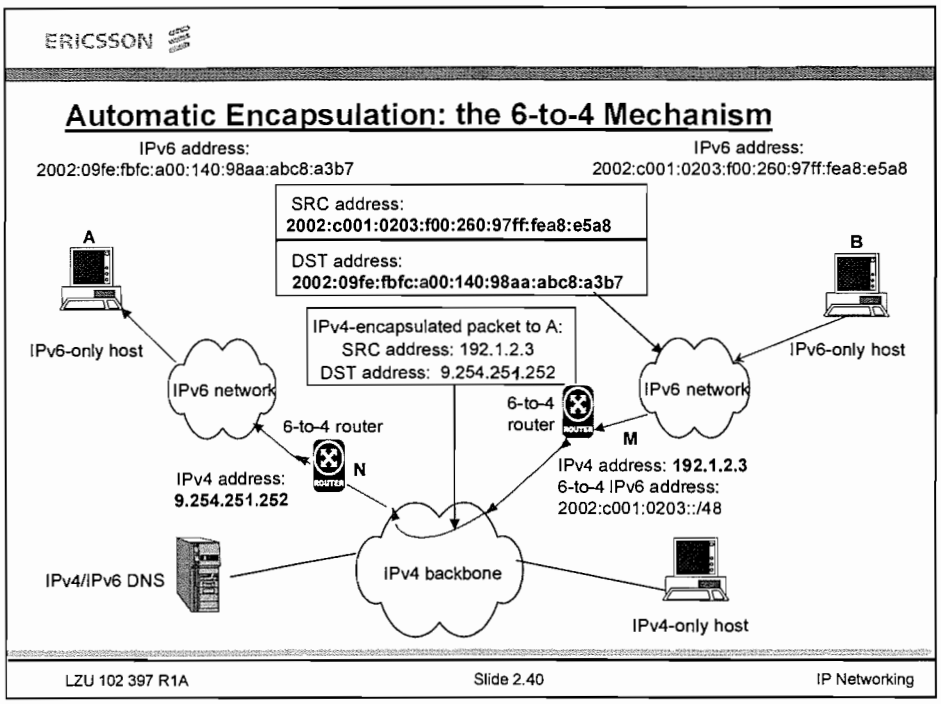


Figure 2-39.

Notes:

Stateless IP/ ICMP Translation Algorithm

The Stateless IP/ICMP Translation Algorithm (SIIT) specification does not cover how an IPv6 node can acquire a temporary IPv4 address from the pool of IPv4 addresses, or how such a temporary address is registered in the DNS.

The temporary IPv4 address is used as an IPv4-translated IPv6 address and the packets travel through a stateless IP/ICMP translator. The IP/ICMP translator translates the packet headers between IPv4 and IPv6. It also translates the addresses in those headers between IPv4 addresses on one side and IPv4-translated IPv6 addresses or IPv4-mapped IPv6 addresses on the other side.

When the IPv4-to-IPv6 translator receives an IPv4 datagram addressed to a destination that lies outside of the attached IPv4 island, it translates the IPv4 header of that packet into an IPv6 header.

It then forwards the packet based on the IPv6 destination address. The original IPv4 header on the packet is removed and replaced by an IPv6 header.

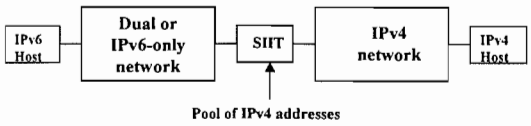
For ICMP messages, all packets need to have the Type value translated, and for ICMP error messages the included IP header also needs translation. When the IPv6-to-IPv4 translator receives an IPv6 datagram addressed to an IPv4-mapped IPv6 address, it translates the IPv6 header of that packet into an IPv4 header. It then forwards the packet based on the IPv4 destination address.

The original IPv6 header on the packet is removed and replaced by an IPv4 header. As before, for ICMP messages, all packets need to have the Type value translated, and for ICMP error messages, the included IP header also needs translation.

ERICSSON

Stateless IP/ICMP Translation Algorithm

- SIIT allows communication between IPv6-only and IPv4-only nodes.
- SIIT translates IPv4 headers to IPv6 headers and vice versa, and some of the ICMP messages between devices with different versions of IP.
- SIIT uses the the IPv4-to-IPv6 translator to form the source and destination addresses.
- The translator is assumed to know the IPv4 addresses that represent the internal IPv6-only nodes. If the IPv4 destination field contains an address that falls in these configured sets of prefixes, the packet must be translated to IPv6.



LZU 102 397 R1A Slide 2.41 IP Networking

Figure 2-40.

Notes:

Network Address Translation – Protocol Translation

Network Address Translation - Protocol Translation (NAT-PT) uses a pool of IPv4 addresses. These IPv4 addresses are assigned to IPv6 nodes on a dynamic basis as sessions are initiated across IPv4-IPv6 boundaries.

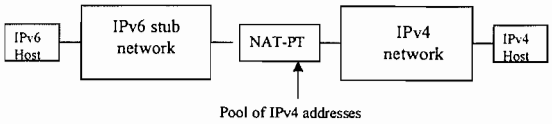
The IPv4 addresses are assumed to be globally unique. NAT-PT binds addresses in the IPv6 network with addresses in the IPv4 network, and vice versa, to provide transparent routing. This binding requires no changes to end nodes and IP packet routing is completely transparent to end nodes. It does require NAT-PT to track the sessions it supports, and obliges inbound and outbound datagrams pertaining to a session to traverse the same NAT-PT router.

The topology restrictions on NAT-PT are the same as those for IPv4 NATs. NAT-PT provides a complete solution that allows a large number of applications to inter-operate between IPv6-only nodes and IPv4-only nodes, without requiring any changes to these applications.

ERICSSON

Network Address Translation - Protocol Translation

- NAT-PT assigns globally unique v4 addresses to v6 nodes on a dynamic basis.
- The topology restrictions on NAT-PT are the same as those on NATv4.
- Combining SIIT with NAT-PT allows applications to interoperate between IPv6-only nodes and IPv4-only nodes without any change in the application.



- IPv6 hosts address IPv4 hosts behind the IPv4 network with a PREFIX::/96, where PREFIX is advertised by the NAT-PT.
- Any returning traffic is recognised as belonging to the same session.
- NAT-PT operation is limited to packets which use TCP.

LZU 102 397 R1A Slide 2.42 IP Networking

Figure 2-41.

Notes:

Intentionally Blank

3 TCP & UDP

After completing this chapter you will be able to:

- Describe TCP Transport Protocol
- Describe UDP Transport Protocol
- Describe the operation of ARP
- Describe the operation of BOOTP / DHCP
- Describe the operation of DNS
- Describe the operation of Traceroute

Intentionally Blank

TRANSMISSION CONTROL PROTOCOL

TCP is a reliable, connection-oriented delivery service. Connection-oriented means that a session must be established before devices can exchange data. There are exactly two endpoints communicating with each other on a TCP connection.

Broadcasting and multicasting are not applicable to TCP. Processes or applications communicate with each other by having both the sending and receiving device create end points, called sockets. A socket consists of the IP address of the device and a 16-bit number called a port. A port is used by transport protocols to identify to which application protocol or process they must deliver incoming messages. Two sockets are required for a virtual connection.

TCP views the data stream as a sequence of octets or bytes that is divided into segments for transmission. Each segment travels across the network in a single IP packet. Reliability is achieved by assigning a sequence number to each segment. When TCP sends a segment it maintains a timer, waiting for the other end to acknowledge reception of the segment. If an acknowledgement is not received within the timer period, the segment is retransmitted.

TCP also provides flow control. Each end of a TCP connection has a finite amount of buffer space. A receiving TCP only allows the other end to send as much data as the receiver has buffers for. This prevents a fast host from taking all the buffers from a slower host.

TCP also reacts to congestion on the network and automatically adjusts the transmission speed to the bandwidth available on the network.

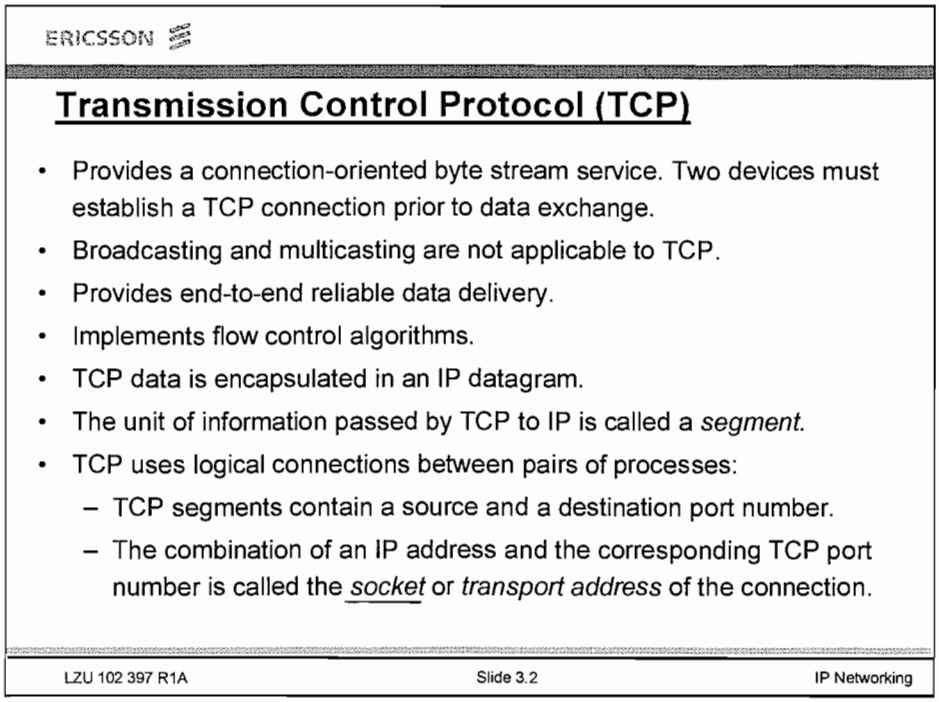


Figure 3-1.

Notes:

Transmission Control Protocol (cont)

A TCP session is initialised by means of a three-way handshake. During this process, the two communicating devices synchronise the sending and receiving of segments, inform each other of the amount of data they are able to receive at once (window size and segment size), and establish a virtual connection.

TCP advertises a window size during connection establishment. The communicating ends set the buffer for the connection by looking at the window size advertised by the other end.

Since a TCP connection is full duplex - data can flow in each direction independently of the other direction - each direction must be closed independently. Each end signals to the other end that it wants to close a connection by setting the FIN flag in the segment.

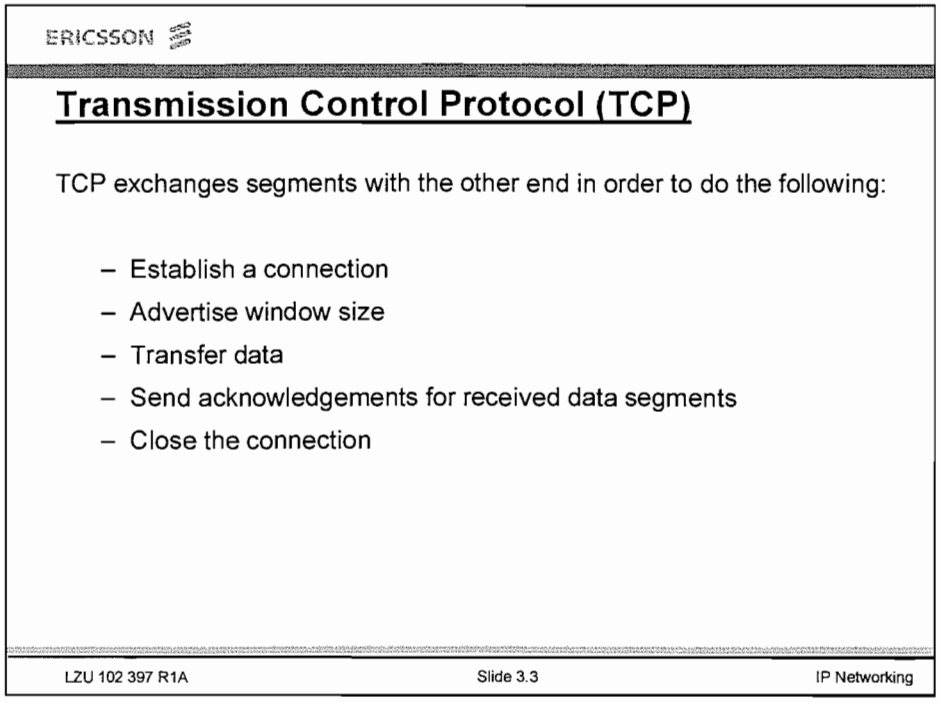


Figure 3-2.

Notes:

TCP Packet Header

The header of a TCP packet is as shown. The function of each of the components is as follows:

Header Element	Number of bits	Meaning
Source port	16	The TCP port number of the sending device.
Destination port	16	The TCP port number of the receiving device.
Sequence number	32	The sequence number of the data byte stream in the segment.
Acknowledgement number	32	The sequence number that the receiver expects to receive next.
Offset	4	The number of 32-bit words in the TCP header. It is needed because the Options field length is variable
Reserved	6	Reserved for future use. Must be zero.
Flags	8	These are six flags that control the behaviour of a TCP packet. They are: 1. Urgent (URG), 2. Acknowledgement (ACK), 3. Push (PSH), 4. Reset connection (RST), 5. Synchronous (SYN), 6. Finish (FIN).
Window	16	Used in acknowledgement segments to implement flow control. Specifies the number of data bytes, which the receiver's buffer can accept.
Checksum	16	Used to verify the integrity of the TCP header and data. The checksum is performed on a 96 bit pseudo header conceptually prefixed to the TCP header, (Source Address, Destination Address, the Protocol, and TCP length.), as well as TCP header and TCP data.
Urgent Pointer	16	When urgent data is being sent (as specified in the code bits), this points to the end of the urgent data in the segment.
Options	Variable	The most common option field is the maximum segment size (MSS) option. It specifies the maximum sized segment that the sender wants to receive.

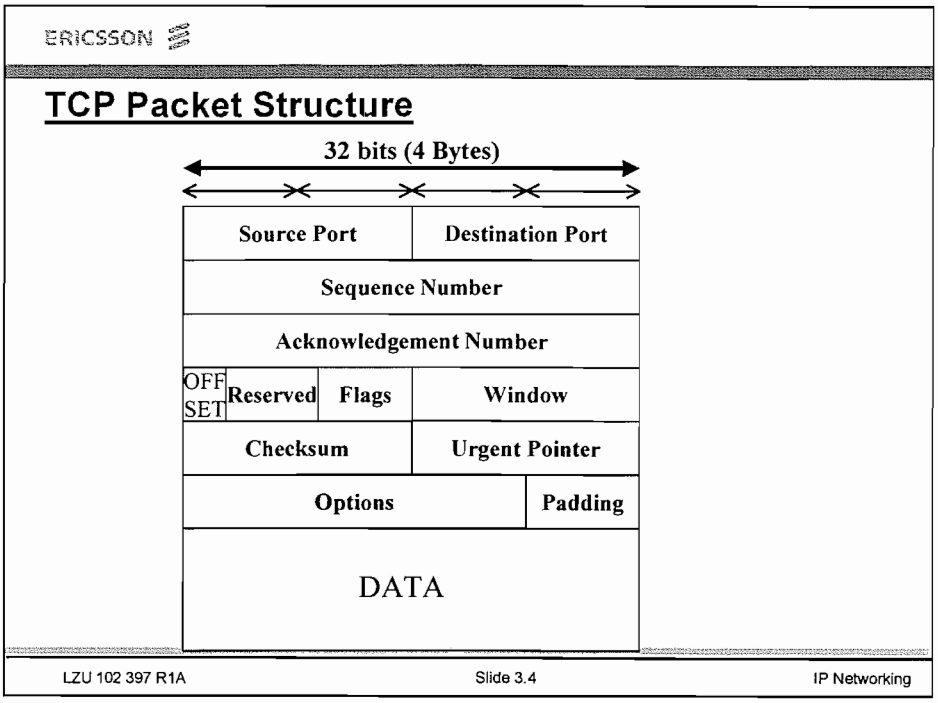


Figure 3-3.

Notes:

TCP Flags

The SYN flag is set during the three-way handshake connection establishment.

The FIN flag is set during connection termination. The receipt of a FIN segment means there will be no more data flowing in that direction. The FIN segment must be acknowledged and the connection is closed on that direction.

The ACK flag is set when the receiving TCP generates a segment to acknowledge the received data sent by the sender.

The sender sets the Push flag to instruct the receiver to pass all its data to the receiving process.

A common situation where a reset is generated is when a connection request arrives and no process is listening on the destination port. Another is when one end is willing to immediately abort the connection. That end transmits a segment with the reset flag set.

TCP provides a so-called "urgent mode", allowing one end to tell the other end that urgent data has been placed into the normal stream of data. The other end is notified that this urgent data has been placed into the data stream and it is up to the receiving end to decide what to do. The urgent pointer points to the last byte of urgent data.

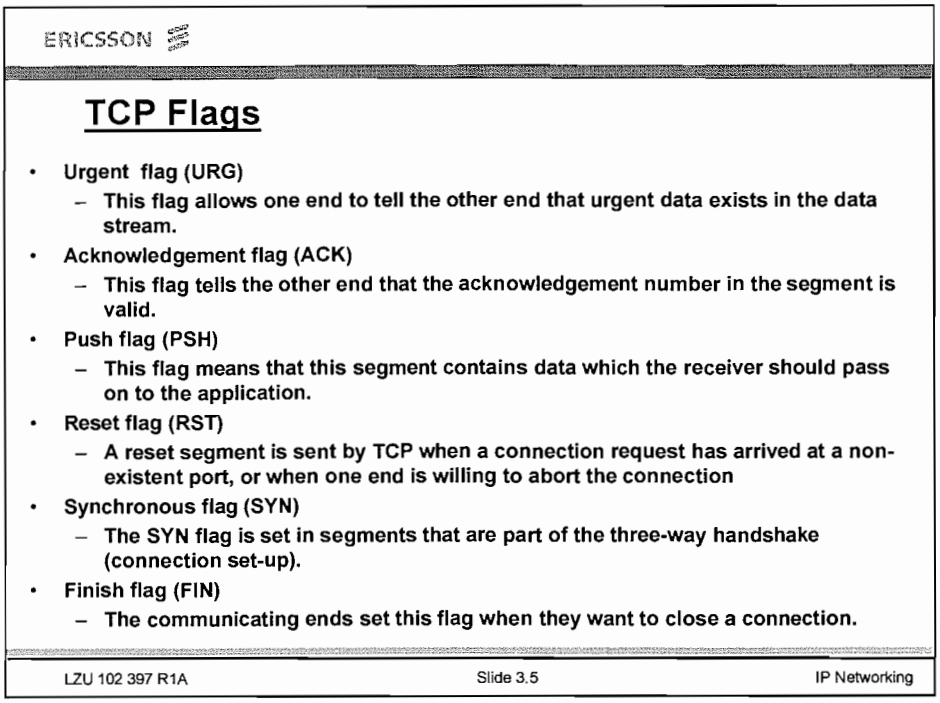


Figure 3-4.

Notes:

Port Numbers

To allow for many processes within a single host to use TCP communication facilities simultaneously, TCP provides a set of addresses or ports within each host.

The port concatenated with the host address from the IP layer forms a socket. A pair of sockets uniquely identifies each connection.

The concatenation of destination IP address and destination port may be used simultaneously in multiple connections. However the concatenation of source IP address and port will differ for each individual connection. Note that there may be multiple connections between the same end points, but that the source port will differ for each connection.

The binding of ports to processes is handled independently by each host. However, TCP combines static and dynamic port binding, using a set of well-known port numbers for commonly invoked programs (for example, e-mail), but leaving most port numbers available for the operating system to allocate, as programs need them. These services can then be accessed through the known port addresses.

Establishing and learning the port addresses of other processes may involve more dynamic mechanisms.

Every TCP segment contains the source and destination port number in order to identify the sending and receiving applications.

A port can use any number between 0 and 65,536. All well-known port numbers are below 1024. Port numbers over 1024 have now been assigned. The diagram opposite lists some well-known ports.

See http://www.iana.org/assignments/port-numbers for up to date port assignments

Network Address Translators (NAT) change the source IP address and source port of a client. This also requires the NAT device to recalculate the TCP checksum within the TCP header.

Network Address Port Translators (NAPT) change the source port selected by a client. This also requires the NAPT device to recalculate the TCP checksum within the TCP header.

ERICSSON

Well-Known Port Numbers

Port Number	Description
7	Echo
20	File Transfer Protocol (FTP) data
21	File Transfer Protocol (FTP) control
22	SSH Remote Login Protocol (SSH)
23	Telnet
25	Simple Mail Transfer Protocol (SMTP)
53	Domain Name Server (DNS)
79	Finger
80	World Wide Web (WWW)
110	Post Office Protocol (POP3)
143	Internet Message Access Protocol (IMAP)
161	SNMP
162	SNMP Traps
~~1080~~	~~Socks~~

LZU 102 397 R1A Slide 3.6 IP Networking

Figure 3-5.

Notes:

Establishing a TCP Connection

In order to establish a connection, TCP uses a three-way handshake.

The client's TCP software generates a sequence number (1000 in the example opposite). The client requests a session by sending out a segment with the synchronisation (SYN) flag set to on. The segment header also includes the sequence number, the size of its receive buffer (window size) and the size of the biggest data segment that it can handle.

The side that sends the first SYN is said to perform an active open. The server acknowledges (ACK) the request by sending back a segment with the synchronisation (SYN) flag set to on. The segment header contains the server's own start-up sequence number and the acknowledgement for the previous SYN segment it received from the client. The segment header also includes the size of the server's receive buffer (window size) and the size of the biggest data segment it can handle.

The client then sends back an acknowledgement of the server's start-up sequence segment. This is the third and last segment from the three-way handshake.

The purpose of these sequence numbers is to prevent delayed packets from being delivered later and then misinterpreted as part of an existing connection.

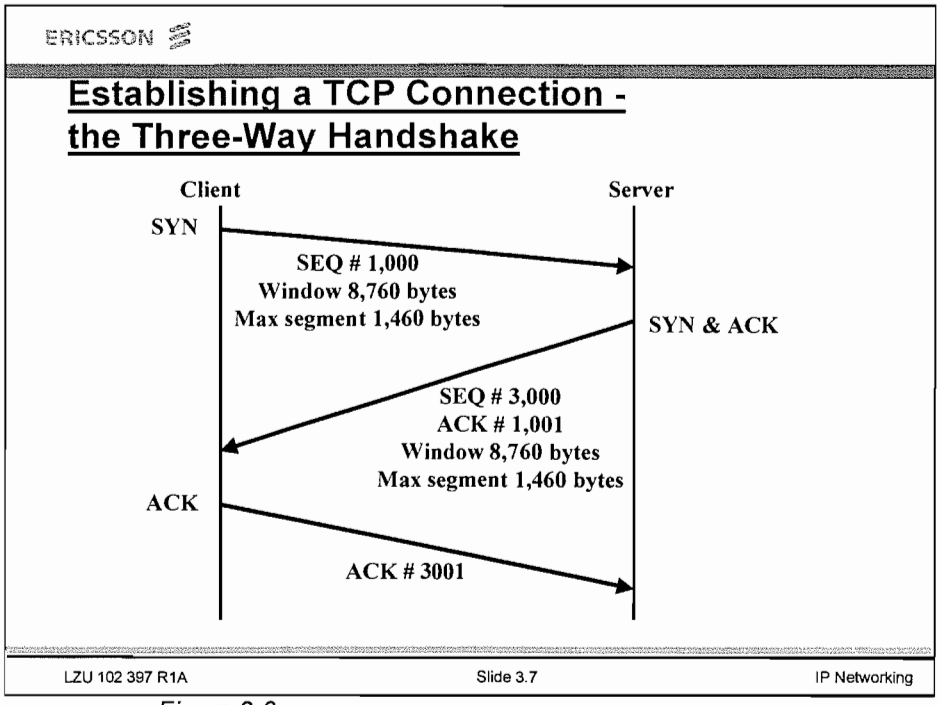


Figure 3-6.

Notes:

Closing a TCP Connection

While it takes three segments to establish a connection, it takes four to terminate a connection.

This is caused by TCP's 'half close'. Since a TCP connection is full-duplex, each direction must be shut down independently. The rule is that either end can send a FIN when it is finished sending data.

When a TCP receives a FIN, it must notify the application that the other end has terminated that direction of data flow. The sending of a FIN is normally the result of the application issuing a close. The receipt of a FIN only means that there will be no more data flowing in that direction.

A TCP can still send data after receiving a FIN (the other direction is still active). While it is possible for an application to take advantage of this half-close, in practice very few TCP applications use it.

The end that first issues the close (sends the first FIN) performs the active close and the other end (that receives the FIN) performs the passive close. When one end receives the FIN, it sends back an ACK of the received sequence number plus one. A FIN consumes a sequence number, just like a SYN.

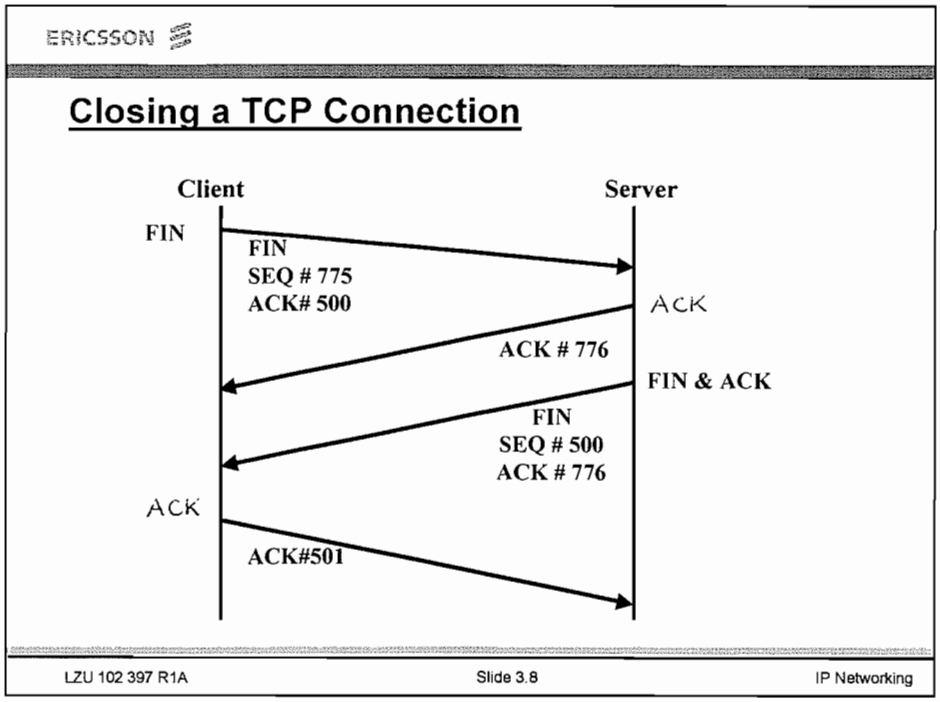


Figure 3-7.

Notes:

Sliding Window Protocol

The Sliding Window Protocol operates over the buffer space allocated for a connection. This buffer space is divided into segments of the length of maximum segment size (MSS).

With the Sliding Window Protocol, the sender can limit the number of outstanding segments (segments sent by the sender but not acknowledged by the receiver) in the network. That way, the rate at which the sender can send data over the network has an upper limit, depending on the window size advertised by the receiver and the round trip time (RTT) of the connection.

Over time this sliding window moves to the right as the receiver acknowledges data. Three terms are used to describe the movement of the right and left edges of the window: closing, opening, and shrinking.

The window closes as the left edge advances to the right. This happens when data is sent. The window opens when the right edge moves to the right, when acknowledgements are received from the other end, allowing more data to be sent. This happens when the receiving processes on the other end reads data and sends acknowledgements, freeing up space in its TCP receive buffer.

The window shrinks when the right edge moves over the left edge. This is strongly discouraged but TCP must be able to cope with this. The left edge of the window cannot move to the left, because the acknowledgement number received from the other end controls this edge. If an ACK that implies moving the left edge to the left is received, it is a delayed duplicate ACK and is silently discarded.

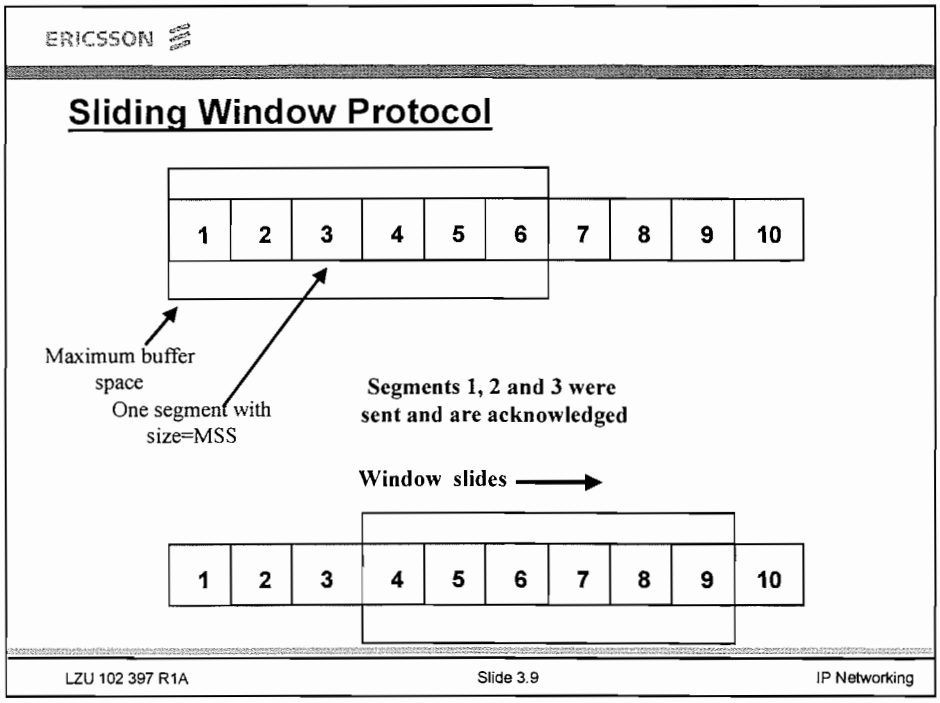


Figure 3-8.

Notes:

Timeout and Retransmission

TCP provides a reliable transport layer. It provides reliability by acknowledging the data it receives from the other end.

However, data segments and acknowledgements can get lost. TCP handles this loss by setting a timeout with the data. If the data is not acknowledged when the timeout expires, TCP retransmits the data.

Fundamental to TCP's timeout and retransmission is the measurement of the round-trip time (RTT) experienced on a given connection. This can change over time as the routes the packets follow can change, as can the network traffic.

TCP should track these changes and modify its timeout accordingly. The more exact estimation of RTT is beneficial because TCP should retransmit a lost packet as soon as possible, but it should not retransmit it if the packet is merely delayed in its way to the destination.

ERICSSON

Timeout and Retransmission

- TCP provides reliability by acknowledging the data it receives.
- Data segments and acknowledgements can get lost in the network.
- TCP measures the round trip time (RTT) of a segment. RTT is the time between sending a data segment and receiving the acknowledgement for it.
- If an ACK is not received in the timeout interval, the segment is retransmitted.
- TCP uses the RTT measurements to estimate the retransmission timeout of the next data segment it transmits.

  $RTT_n = \alpha RTT_{n-1} + (1-\alpha)M$, where -$\alpha$ is a smoothing factor (≈ 0.9) and -M is the new measurement value
- The new retransmission timeout (RTO) interval becomes:

  $RTO = RTT * \beta$, with β a value around 2

LZU 102 397 R1A Slide 3.10 IP Networking

Figure 3-9.

Notes:

Retransmission due to Timeout

In the diagram opposite, when the Sender sends segment 1, timeout for the segment is set. If the data segment is lost, an acknowledgement is not received.

The sender waits for the timer to expire and then resends the segment. If the acknowledgement for the segment is received within the timeout interval, the sender slides its window, sends the next segment, and resets the timer.

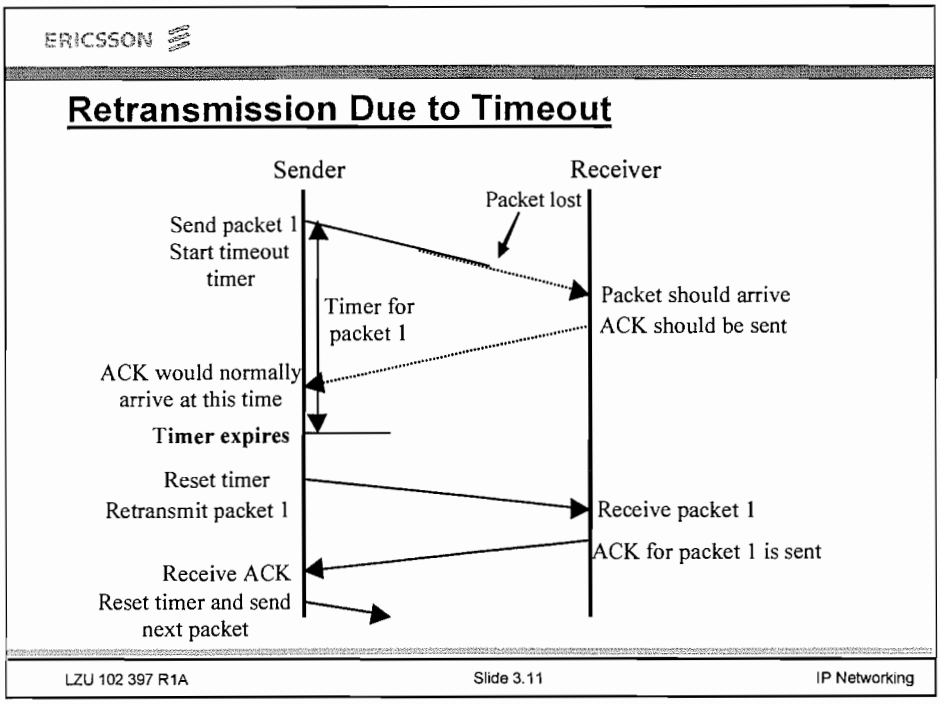


Figure 3-10.

Notes:

Slow Start Algorithm

Old TCPs would start a connection with the sender injecting multiple segments into the network, up to the window size advertised by the receiver.

If there are routers and slower links between the sender and the receiver, problems can arise. Some intermediate router must queue the packets, and it is possible for that router to run out of buffer space and be forced to drop packets. Dropping packets causes performance degradation for TCP.

The Slow Start Algorithm operates by observing that the rate at which new packets should be injected into the network is the rate at which the acknowledgements are returned by the other end.

Slow start adds another window to the sender's TCP, that is, the congestion window, called 'cwnd'. When a new connection is established, the congestion window is initialised to one segment. Each time an ACK is received, the congestion window is increased by one segment.

The sender can transmit up to the minimum of the congestion window and the advertised window. The congestion window is flow control imposed by the sender, while the advertised window is flow control imposed by the receiver.

The sender starts by transmitting one segment and waiting for its ACK. When that ACK is received, the congestion window is incremented from one to two, and two segments can be sent. When each of those two segments is acknowledged, the congestion window is increased to four.

This provides an exponential growth for cwnd. At some point, the capacity of the network will be reached, and an intermediate router will start discarding packets. This tells the sender that its congestion window has become too large.

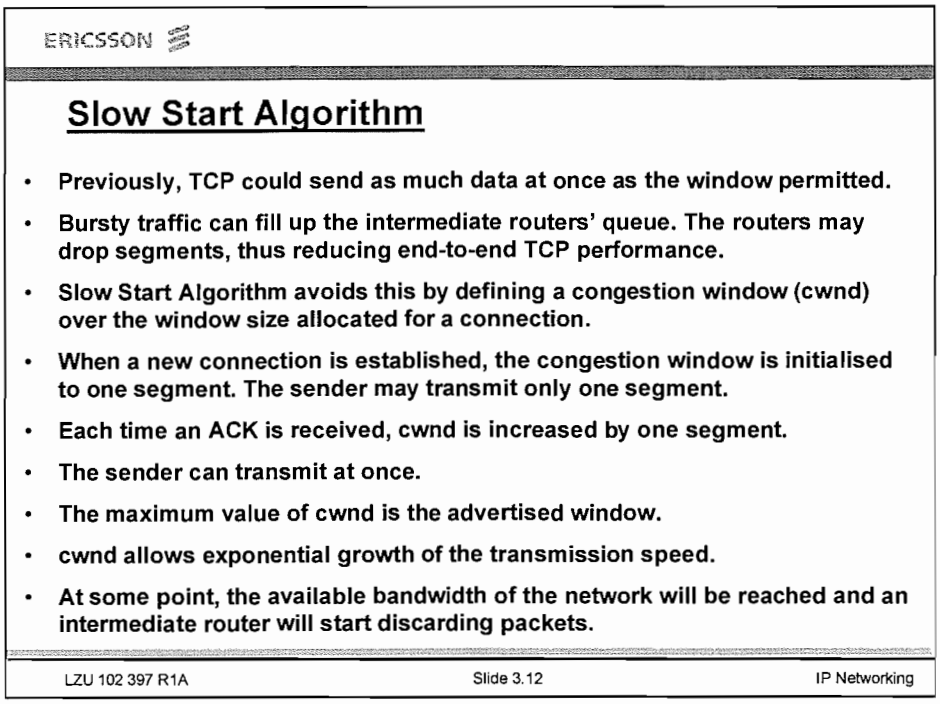


Figure 3-11.

Notes:

Duplicate Acknowledgements

The steps in duplicate acknowledgement are illustrated and explained in the diagram opposite.

First, if an acknowledgment (ACK) for a given segment is not received in a certain amount of time a retransmission timeout occurs and the segment is resent. Second, the "Fast Retransmit" algorithm resends a segment when three duplicate ACKs arrive at the sender. However, because duplicate ACKs from the receiver are also triggered by packet reordering in the Internet, the TCP sender waits for three duplicate ACKs in an attempt to disambiguate segment loss from packet reordering. Once in a loss recovery phase, a number of techniques can be used to retransmit lost segments, including slow start-based recovery or Fast Recovery.

TCP's retransmission timeout is based on measured round-trip times between the sender and receiver. To prevent spurious retransmissions of segments that are only delayed and not lost, the minimum RTO is conservatively chosen to be one second. Therefore, TCP senders have to detect and recover from as many losses as possible without incurring a lengthy timeout when the connection remains idle. However, if not enough duplicate ACKs arrive from the receiver, the Fast Retransmit algorithm is never triggered. This situation occurs when the congestion window is small or if a large number of segments in a window are lost.

For instance, consider a congestion window (cwnd) of three segments. If the network drops one segment, then at most two duplicate ACKs will arrive at the sender. Since three duplicate ACKs are required to trigger Fast Retransmit, a timeout will be required to resend the dropped packet.

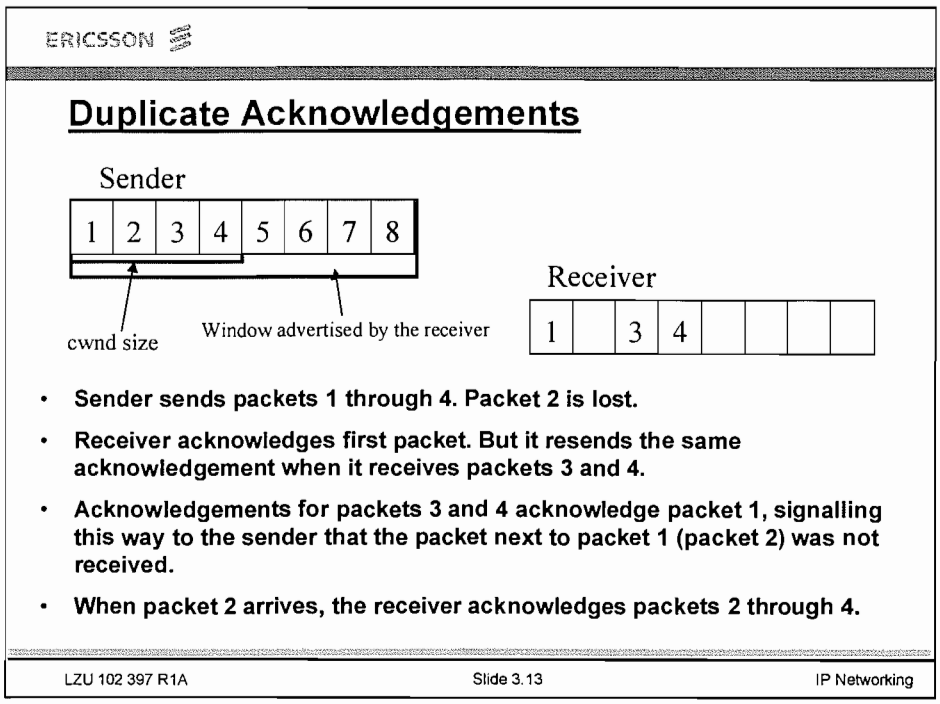


Figure 3-12.

Notes:

Congestion Avoidance Algorithm

Congestion avoidance and slow start require that two variables be maintained for each connection: a congestion window, cwnd, and a slow start threshold size, ssthresh. The combined algorithm operates as follows.

- Initialisation for a given connection sets cwnd to one segment and ssthresh to 65535 bytes.
- When congestion occurs (indicated by a timeout or the reception of duplicate ACKs), one half of the current window size (the minimum of cwnd and the receiver's advertised window, but at least two segments) is saved in ssthresh. Additionally, if the congestion is indicated by a timeout, cwnd is set to one segment (that is, slow start).
- When new data is acknowledged by the other end, cwnd is increased, but the way it increases depends on whether TCP is performing slow start or congestion avoidance. If cwnd is less than or equal to ssthresh, TCP is in slow start; otherwise, TCP is performing congestion avoidance. Slow start continues until TCP is halfway to where it was when congestion occurred, and then congestion avoidance takes over. With slow start, cwnd begins at one segment, and is incremented by one segment every time an ACK is received.
- This opens the window exponentially, that is, one segment is sent, then two, then four, and so on. Congestion avoidance dictates that cwnd be incremented by segsize*segsize/cwnd each time an ACK is received, where segsize is the segment size and cwnd is maintained in bytes. This is a linear growth of cwnd, compared to slow start's exponential growth. The increase in cwnd should be at most one segment each round-trip time, whereas slow start increments cwnd by the number of ACKs received in a round-trip time.

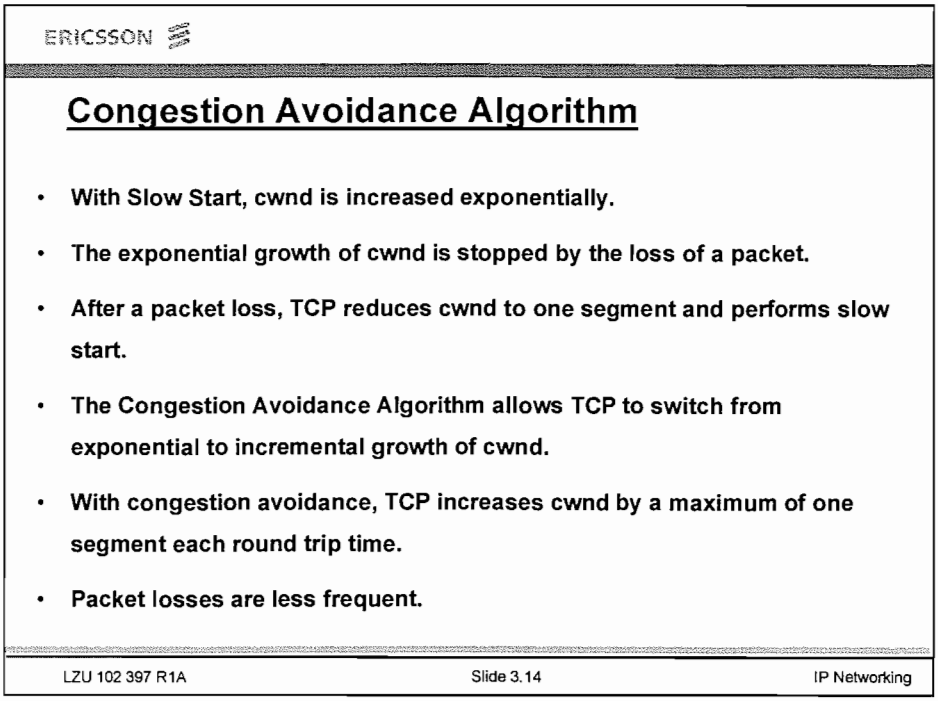


Figure 3-13.

Notes:

Fast Retransmit Algorithm

TCP does not know whether a duplicate ACK is caused by a lost segment or merely by a reordering of segments. TCP must therefore wait for a small number of duplicate ACKs to be received.

It is assumed that if there is just a reordering of the segments, there will be only one or two duplicate ACKs before the reordered segment is processed. A new ACK is then generated.

If three or more duplicate ACKs are received in a row, it is a strong indication that a segment has been lost. TCP then retransmits what appears to be the missing segment, without waiting for a retransmission timer to expire.

Larger TCP initial windows would not dramatically increase the burstiness of TCP traffic in the Internet today, because such traffic is already fairly bursty. Bursts of two and three segments are already typical of TCP.

A delayed ACK (covering two previously unacknowledged segments) received during congestion avoidance causes the congestion window to slide and two segments to be sent. The same delayed ACK received during slow start causes the window to slide by two segments and then be incremented by one segment, resulting in a three-segment burst.

While not necessarily typical, bursts of four and five segments for TCP are not rare. Assuming delayed ACKs, a single dropped ACK causes the subsequent ACK to cover four previously unacknowledged segments. During congestion avoidance this leads to a four-segment burst, and during slow start a five-segment burst is generated.

ERICSSON

Fast Retransmit and Fast Recovery Algorithms

- If three or more duplicate ACKs are received in a row, a segment has probably been lost.
- TCP then retransmits what appears to be the missing segment, without waiting for a retransmission timer to expire.
- cwnd is reduced to half of its initial value, not to one as in case of timeout. This is the Fast Retransmit Algorithm.
- After fast retransmit sends what appears to be the missing segment, congestion avoidance, but not slow start is performed.
- It is an improvement that allows high throughput under moderate congestion, especially for large windows. This is Fast Recovery.
- TCP supposes that one segment can be lost accidentally on the network, but more than one loss in a row indicates congestion.
- TCP never does Fast Retransmit for more than one segment from the same congestion window of the sender.

LZU 102 397 R1A Slide 3.15 IP Networking

Figure 3-14.

Notes:

Fast Recovery Algorithm

After fast retransmit sends what appears to be the missing segment, congestion avoidance, but not slow start, is performed. This is the Fast Recovery Algorithm. It allows high throughput under moderate congestion, especially for large windows.

The reason for not performing slow start in this case is that receiving duplicate ACKs tells TCP more than just a packet has been lost.

Since the receiver can only generate the duplicate ACK when another segment is received, that segment has left the network and is in the receiver's buffer. That is, data is still flowing between the two ends, and TCP does not want to reduce the flow abruptly by going into slow start.

The Fast Retransmit and Fast Recovery Algorithms are usually implemented together.

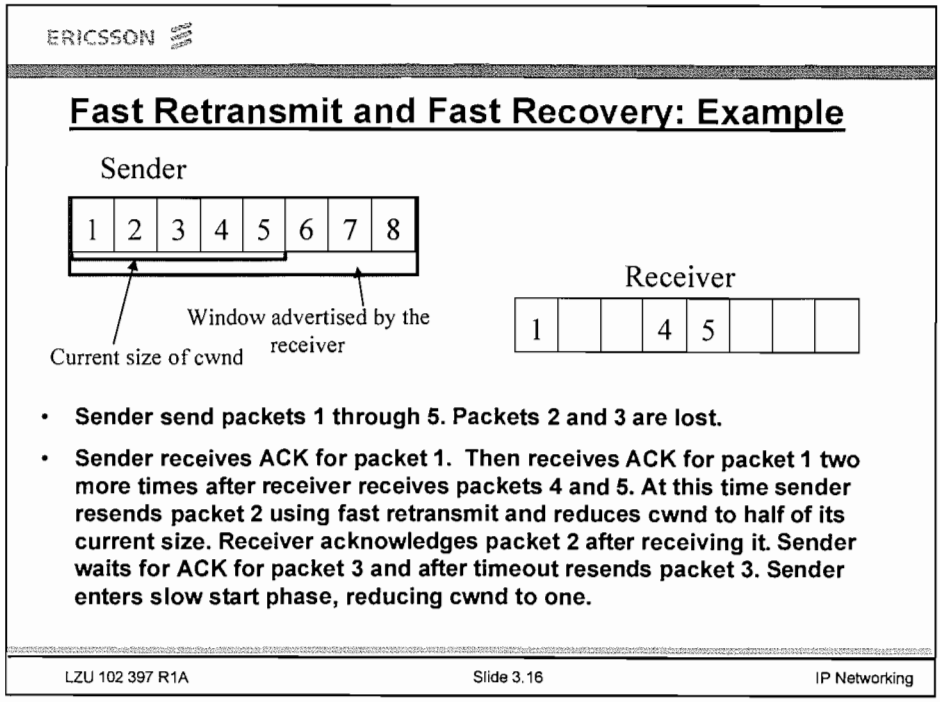


Figure 3-15

Notes:

USER DATAGRAM PROTOCOL (UDP)

User Datagram Protocol (UDP) provides a connectionless packet service that offers unreliable 'best effort' delivery. This means that the arrival of packets is not guaranteed, nor is the correct sequencing of delivered packets.

Applications that do not require an acknowledgement of receipt of data, for example, audio or video broadcasting use UDP.

UDP is also used by applications that typically transmit small amounts of data at one time, for example, the Simple Network Management Protocol (SNMP).

UDP provides a mechanism that application programs use to send data to other application programs. UDP provides protocol port numbers used to distinguish between multiple programs executing on a single device. That is, in addition to the data sent, each UDP message contains both a destination port number and a source port number. This makes it possible for the UDP software at the destination to deliver the message to the correct application program, and for the application program to send a reply.

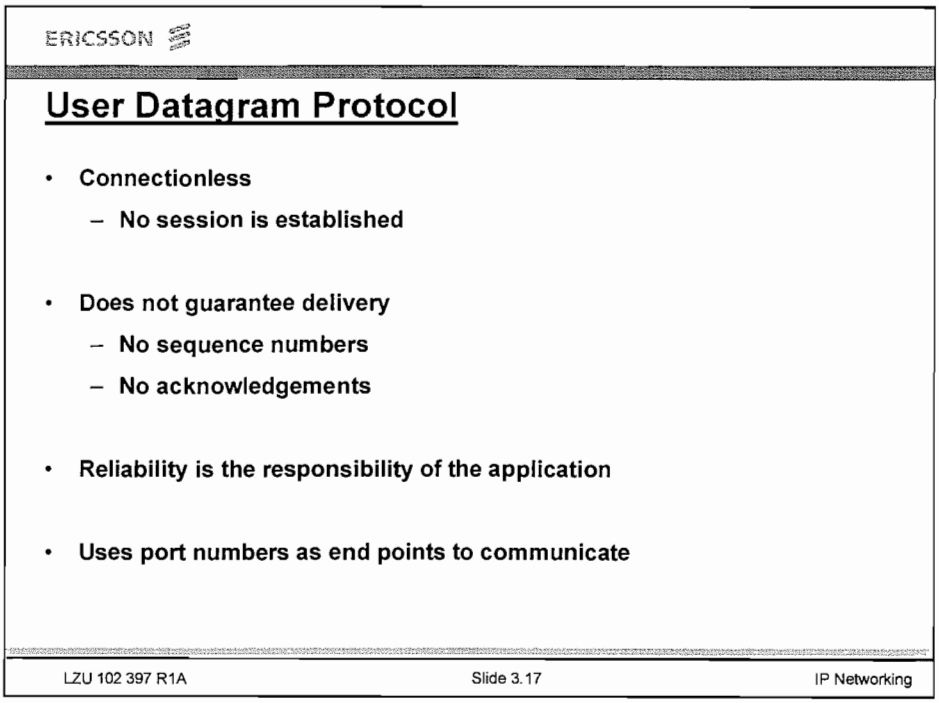


Figure 3-16.

Notes:

The UDP header is divided into the following four 16-bit fields.

Header Element	Purpose
Source port	This is the UDP port number of the process on the sending device.
Destination port	This is the UDP port number of the process on the destination device.
Length	This is the size in bytes of the UDP packet, including the header and data. The minimum length is 8 bytes, the length of the header alone.
UDP Checksum	This is used to verify the integrity of the UDP header. The checksum is performed on a "pseudo header" consisting of information obtained from the IP header (source and destination addresses, and protocol number) as well as the UDP header.

The purpose of using a pseudo-header is to verify that the UDP packet has reached its correct destination. The correct destination consists of a specific machine and a specific protocol port number within that machine.

The UDP header itself specifies only the protocol port number. Thus, to verify the destination, UDP on the sending machine computes a checksum that covers the destination IP address as well as the UDP packet. At the ultimate destination, UDP software verifies the checksum using the destination IP address obtained from the header of the IP packet that carried the UDP message. If the checksum agrees, then it must be true that the packet has reached the intended destination host as well as the correct protocol port within that host.

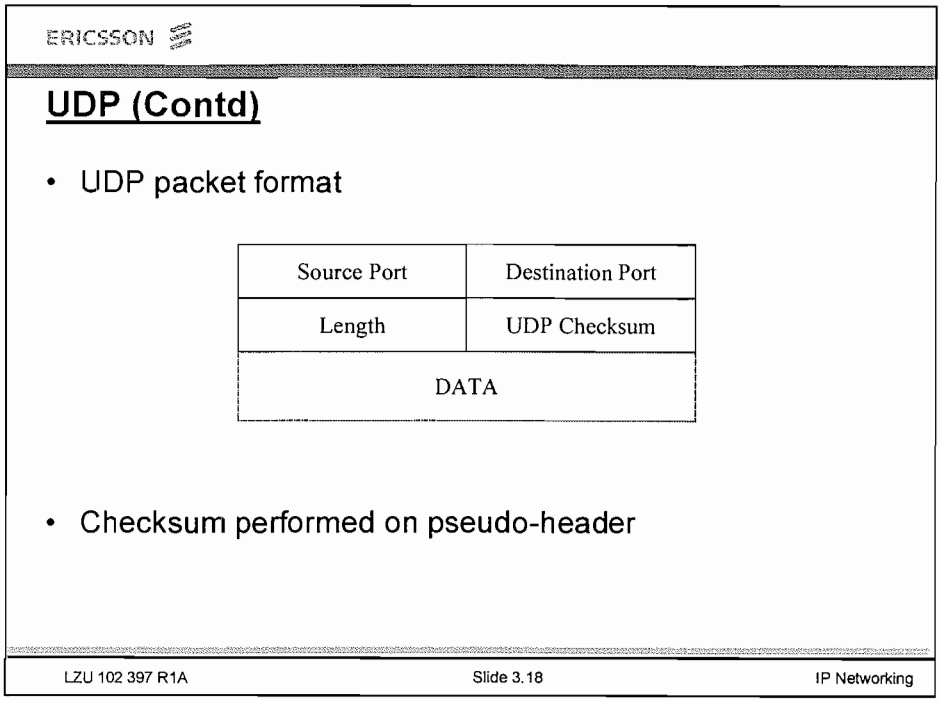


Figure 3-17.

Notes:

ADDRESS RESOLUTION PROTOCOL (ARP)

Network devices must know each other's hardware address (MACS) in order to communicate on an Ethernet network. Address resolution is the process of mapping a host's IP address to its hardware address.

The Address Resolution Protocol (ARP) is responsible for obtaining hardware addresses of TCP/IP devices on broadcast-based networks. ARP uses a local broadcast of the destination IP address to acquire the hardware address of the destination device.

Once the hardware address is obtained, both the IP address and the hardware address are stored as an entry in the ARP cache for a period of time. This is called a dynamic entry.

The ARP cache is always checked for an IP address / hardware address mapping before initiating an ARP request broadcast.

An alternative to dynamic entries is to use static entries. In this case the IP address/hardware address mapping is manually entered into the ARP cache.

Static entries reduce broadcast traffic on the network. The disadvantage of static entries is that they are time consuming to implement and if either the IP or the hardware address of a remote device changes, the entry in the ARP cache will be incorrect and thus prevent the two devices from communicating.

ERICSSON

Address Resolution Protocol (ARP)

- A source must know a destination's hardware address before it can send an IP packet directly to it.
- ARP is the mechanism that maps IP to hardware addresses.
- ARP uses a local broadcast to obtain a hardware address dynamically.
- ARP stores mappings in cache for future use.
- Static entries can be manually entered into the ARP cache.

LZU 102 397 R1A Slide 3.19 IP Networking

Figure 3-18.

Notes:

ARP Request

The source device knows its own IP and hardware address and the IP address of the destination device that it wants to send the information.

It checks its ARP cache for the hardware address of the destination host. If no mapping is found, the source builds an ARP request packet, looking for the hardware address to match the IP address.

The ARP request is a broadcast so all local devices receive and process it. Each device checks for a match with its own IP address.

The destination device determines that there is a match and will send an ARP reply directly to the source device with its hardware address.

Both devices update their ARP cache with the IP address/hardware address mapping of the other device. From then on the devices can communicate directly with each other.

If devices do not communicate with each other after a period of time they will clear the entry from their ARP caches.

Note that if the destination host is on a remote network, the IP software determines the IP address of a locally attached next-hop router to which to send the IP packet. The sending device then uses ARP to obtain the hardware address of the local router (not the remote destination host)

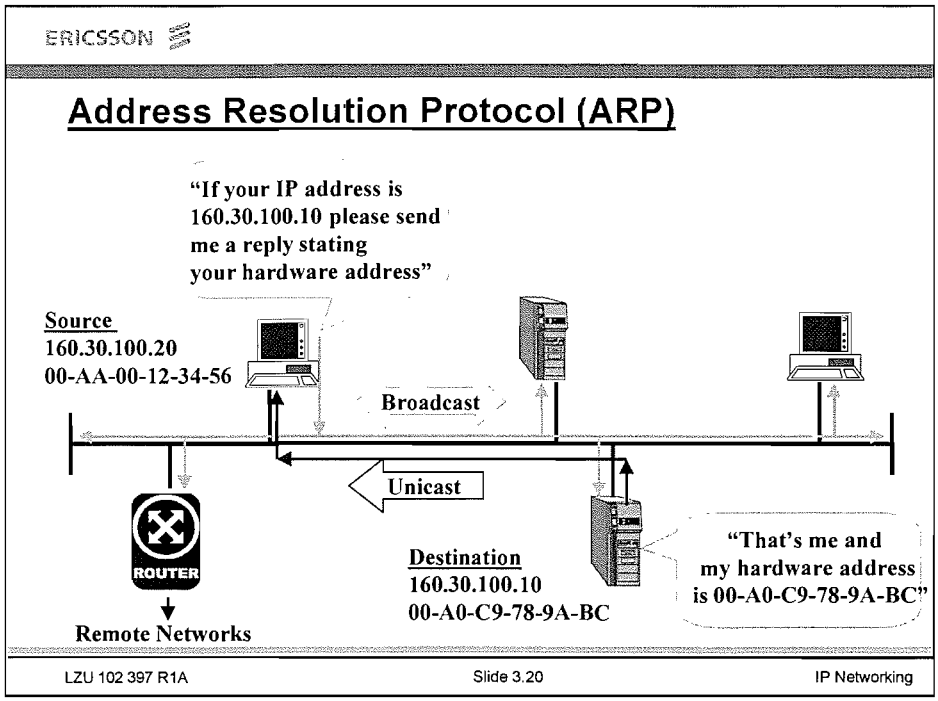


Figure 3-19.

Notes:

ARP Packet Structure

The following table describes the elements of an ARP packet:

Packet Element	Number of Bits	Purpose
Hardware Type	16	This specifies the hardware interface type, for example, Ethernet has a value of 1.
Protocol Type	16	This specifies the higher-level protocol whose address needs to be mapped onto the hardware, for example, IP - 0800.
HLEN, hardware address length	8	This specifies the length in bytes of the hardware address in this packet, for example, Ethernet - 6.
PLEN, protocol address length	8	This specifies the length in bytes of the protocol address in this packet. For IP this is four.
Operation code	16	This specifies whether this is an ARP request - Op code 1 or an ARP reply - Op code 2.
Sender's hardware address	48	This contains the hardware address of the sender (the ARP requester).
Sender's IP address	32	This contains the protocol address of the sender (the ARP requester).
Target's hardware address	48	This contains the hardware address of the target (the ARP responder).
Target's IP address	32	This contains the protocol address of the sender (the ARP responder).

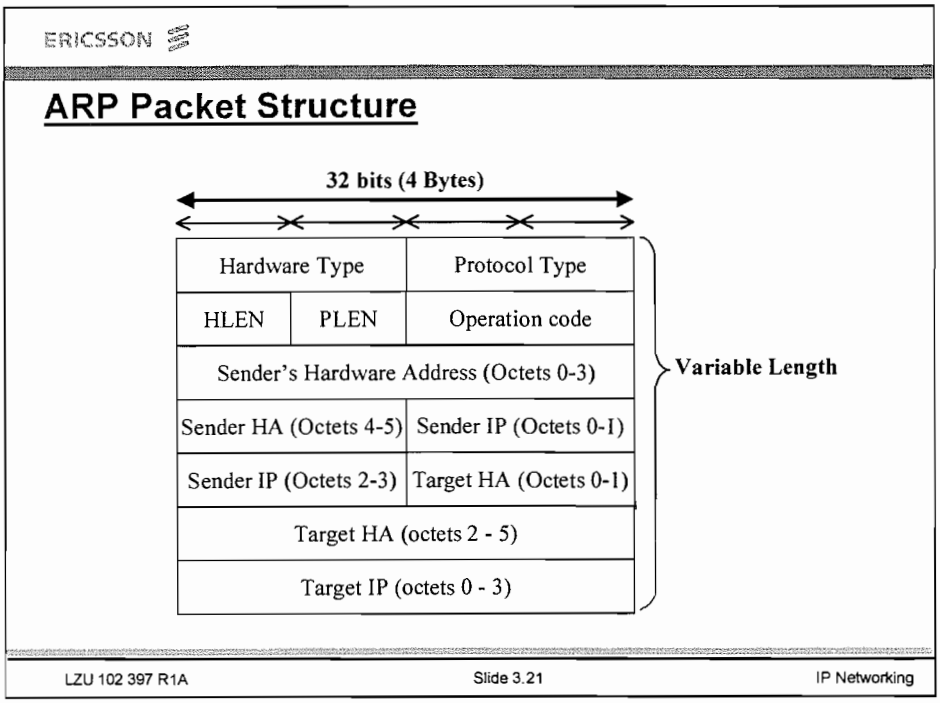


Figure 3-20.

Notes:

REVERSE ARP

ARP solves the problem of mapping a host's IP address to its hardware address, but sometimes the reverse problem must be solved.

Reverse ARP (RARP) is used when the hardware address is given, for example an Ethernet address, but not its corresponding IP address.

The RARP protocol allows a newly booted device to broadcast its Ethernet address and say:

'my 48-bit Ethernet address is 00-A0-C9-78-9A-BC. Does anyone know my IP address?'

The RARP protocol uses the same message format as ARP. The Operation codes are 3 for request Reverse and Op code 4 for reply Reverse [RFC903].

The server sees this request, looks up the Ethernet address in its configuration files and sends back the corresponding IP address.

This type of server is known as an RARP server. To prevent multiple servers from sending a reply simultaneously, thus causing collisions, a primary server may be designated for each host wishing to use RARP. This server replies immediately, and all non-primary servers simply listen and note the time of the request.

If the primary server is unavailable, the originating node will time out and re-broadcasts the RARP request. The non-primary servers respond when they hear a copy of the request within a short time after the original broadcast.

For example, printers use RARP to obtain an IP address.

Note that RARP requests stay within the local LAN, so the servers must also reside there.

ERICSSON

Reverse Address Resolution Protocol

- Reverse ARP is the mechanism that maps hardware addresses to IP addresses.
  - RARP protocol allows a newly booted machine to broadcast its Ethernet address.
  - The RARP server sees this request and sends back the corresponding IP address.
  - Uses Operation codes:
    - 3 - request Reverse
    - 4 - reply Reverse

LZU 102 397 R1A Slide 3.22 IP Networking

Figure 3-21.

Notes:

BOOTP / DHCP

The Bootstrap Protocol (BOOTP) and the Dynamic Host Configuration Protocol (DHCP) are upper layer programs, which automate the initial loading and configuration of hosts.

Instead of using the RARP server to obtain an IP address, a newly booted device may use these programs in order to obtain an IP address, a bootable file address, and configuration information.

These programs may be used on networks that dynamically assign hardware addresses (which preclude using RARP). They also provide a centralised management of IP addresses and configuration files, which eliminates the need for per-host information files.

BOOTP and DHCP are actually the same program with different options. BOOTP enhanced with the new options is called DHCP.

The BOOTP sequence commences with a client, for example a diskless workstation, being booted. The client initiates a BOOTP request with a broadcast address to all stations on the local network. The BOOTP request contains the client's hardware (MAC) address. The BOOTP server receives the BOOTP requests, on UDP port 67.

The server looks up the assigned IP address and puts it in the response message. It also adds the name of the BOOTP server and the name of the appropriate load file that may be executed. Depending on the implementation, it may also add other configuration parameters such as the subnet mask and default gateway.

When the client diskless workstation receives the reply (on UDP port 68) it uses the information supplied by the server to initiate a TFTP get message to the server specified. The response to the TFTP get message is an executable load file.

To keep an implementation as simple as possible, BOOTP messages have fixed length fields, and replies have the same format as request. Field OP specifies whether the message is a request (1) or a reply (2).

ERICSSON

BOOTP (BOOTstrap Protocol)

- A newly booted device may use BOOTP to obtain an IP address, a bootable file address, and configuration information.
  - The client initiates a BOOTP request with a broadcast address to all stations on the local network.
  - The BOOTP server monitors for BOOTP requests (on UDP port 67).
  - The server looks up the assigned IP address and puts it in the response message.
  - It also adds the name of the BOOTP server and the name of the appropriate load file that may be executed.
  - It may also add other configuration parameters such as the subnet mask and default gateway.
  - The client receives the reply (on UDP port 68).
  - The client uses the information supplied by the server to initiate a TFTP 'get' message to the server specified.
  - The response to the TFTP 'get' message is an executable load file.
- DHCP is an enhanced version of BOOTP

LZU 102 397 R1A Slide 3.23 IP Networking

Figure 3-22.

Notes:

BOOTP header

The fields HTYPE and HLEN specify the network hardware type and length of the hardware address (for example, Ethernet has type 1 and address length 6). The client sets the HOPS field to zero, but this is incremented by relay agents. The BOOTP server increments the HOPS count if a relay agent is involved in the request. The TRANSACTION ID field contains an integer that diskless machines use to match responses with requests. The SECONDS field reports the number of seconds since the client started to boot.

The CLIENT IP ADDRESS field and all the fields following it contain the most important information. To allow the greatest flexibility, clients fill in as much information as they know and leave remaining fields set to zero. For example, if a client knows the name or address of a particular server from which it wants information, it can fill in the server IP address or server host name fields. If these fields are nonzero only the server with matching name/address will answer the request; if they are zero, any server that receives the request will reply.

BOOTP can be used from a client that already knows its IP address (for example, to obtain boot file information). A client that knows its IP address places it in the CLIENT IP ADDRESS field; other clients use zero. If the client's IP address is zero in the request, a server returns the client's IP address in the YOUR IP ADDRESS field.

The ROUTER IP ADDRESS field is set to zero by the client (0.0.0.0), and if a router handles the request, it records its address in the field.

The CLIENT HARDWARE ADDRESS field is used by the client for its MAC address. The SERVER HOST NAME field is optional and may be set to zero by the client and server.

The BOOT FILE NAME may be set to zero by the client or optionally set to a generic filename to be booted. For example, 'Unix'. The server will replace the field with a fully qualified path and file name of the appropriate boot file.

The VENDOR-SPECIFIC AREA CONTAINS optional information to be passed from the server to the client. This includes IP addresses of routers, timeservers, and DNS servers.

ERICSSON

BOOTP Message Format

0	8	16	24	31
OP	HTYPE	HLEN	HOPS	
TRANSACTION ID				
SECONDS		UNUSED		
CLIENT IP ADDRESS				
YOUR IP ADDRESS				
SERVER IP ADDRESS				
ROUTER IP ADDRESS				
CLIENT HARDWARE ADDRESS (16 OCTETS)				
SERVER HOST NAME (64 OCTETS)				
BOOT FILE NAME (128 OCTETS)				
VENDOR-SPECIFIC AREA (64 OCTETS)				

LZU 102 397 R1A Slide 3.24 IP Networking

Figure 3-23

Notes:

DHCP

DHCP centralises and manages the allocation of TCP/IP configuration information by automatically assigning IP addresses to devices configured to use DHCP.

Implementing DHCP eliminates some of the configuration problems associated with manually configuring TCP/IP.

Typing in the IP address, subnet mask, or default gateway incorrectly can lead to problems including communication difficulties and network problems due to a duplicate IP address.

Each time a DHCP client starts, it requests IP an address from a DHCP server. When a DHCP server receives a request, it selects IP addressing information from a pool of addresses defined in its database and offers it to the DHCP client.

If the client accepts the offer, the IP addressing information is leased to the client for a specified period of time. In addition, the DHCP server will supply a subnet mask and optional values such as default gateway address, Domain Name Server (DNS) address and WINS (Windows Internet Name Service) address.

Non-DHCP clients still need to be configured manually with static addresses.

If there is no available IP addressing information in the pool to lease to a client, the client cannot initialise TCP/IP.

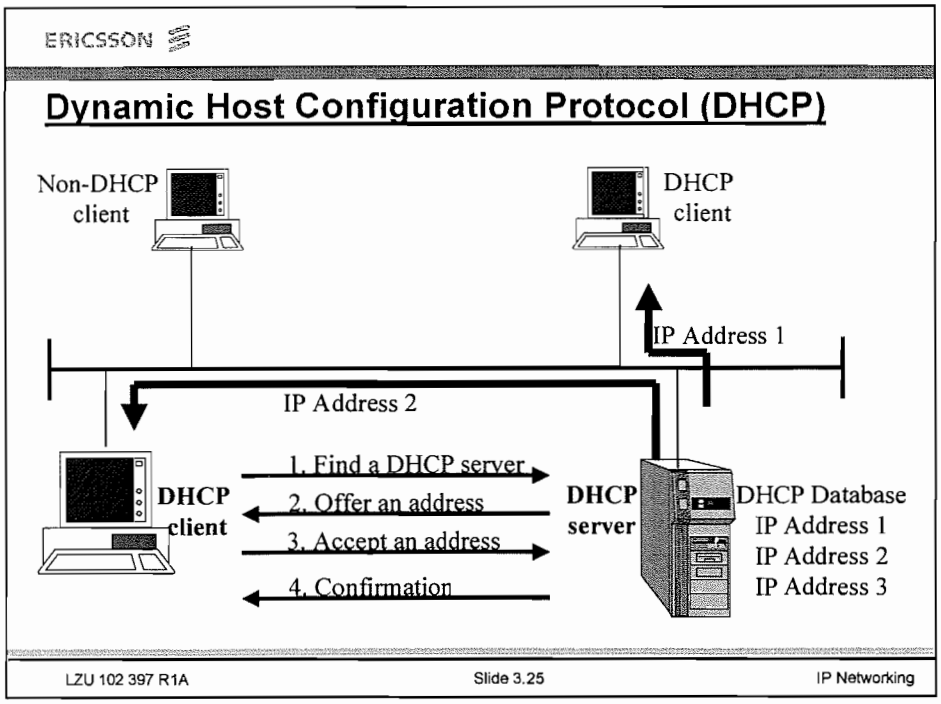


Figure 3-24.

Notes:

DHCP ADDRESS ALLOCATION

DHCP supports three mechanisms for IP address allocation:

Manual Allocation

In this scheme, DHCP is simply used as a mechanism to deliver a predetermined network address and other configuration options to a host. There is a one-to-one mapping between the unique client identifier (generally the Ethernet address) offered by the client during DHCP initialisation and the IP address returned to the client by the DHCP server.

It is necessary for a network administrator to provide the unique client ID/IP address mapping used by the DHCP server.

Automatic Allocation

This is similar to manual allocation in that a permanent mapping exists between a host's unique client identifier and its IP address. However, in automatic allocation this mapping is created during the initial allocation of an IP address.

The IP addresses assigned during automatic allocation come from the same pool as dynamic addresses, but once assigned they cannot be returned to the free address pool without administrative intervention. Both automatic and manually assigned addresses are considered to have permanent leases.

Dynamic Allocation

DHCP assigns an IP address for a limited period of time. This IP address is known as a lease. This mechanism allows addresses that are no longer needed by their host to be automatically re-used.

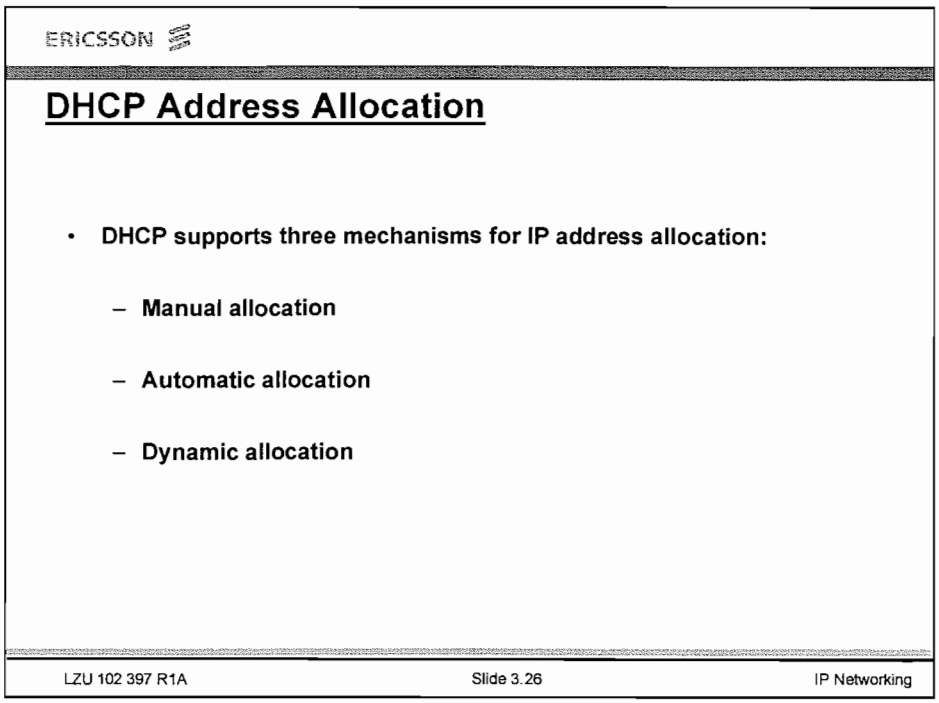


Figure 3-25.

Notes:

DHCP process

DHCP uses a four-phase process to configure a DHCP client. In the first two phases the client sends a discovery message to a DHCP server and a DHCP server offers an IP address to the client. The DHCP client then checks if the offered IP is valid by sending an ARP to that address. If there is no reply to the ARP request, the client then sends a DHCP request to accept the offer and the server acknowledges the allocation

IP Lease Request

The first time a client is initialised, it requests an IP address lease by broadcasting a discovery to all DHCP servers. Because the client does not have an IP address or know the IP address of a DCHP server, it uses 0.0.0.0 as the source address and 255.255.255.255 as the destination address. The discovery message is sent in a DHCPDISCOVER message. This message also contains the client's hardware address and computer name, so those DHCP servers know which client sent the request. The client sets a timer when sending a DHCPDISCOVER message. If it does not receive a response before the timer expires, it resends the DHCPDISCOVER message.

IP Lease Offer

All DHCP servers that receive the request, and have a valid configuration for the client, broadcast an offer with the following information: the client's hardware address, an offered IP address, a subnet mask, the length of the lease and a server identifier (the IP address of the offering DHCP server). A broadcast is used because the client does not yet have an IP address. The offer is sent as a **DHCPOFFER** message. The DHCP server reserves the IP address so that it will not be offered to another DHCP client.

IP Lease Selection

The DHCP client selects the IP address from the first offer it receives. After the client receives an offer from at least one DHCP server, it broadcasts to all DHCP servers that it has made a selection by accepting an offer. The broadcast is sent in a **DHCPREQUEST** message and includes the identifier (IP address) of the server whose offer was accepted. All other DHCP servers then retract their offer so that their IP addresses are available for the next IP lease request

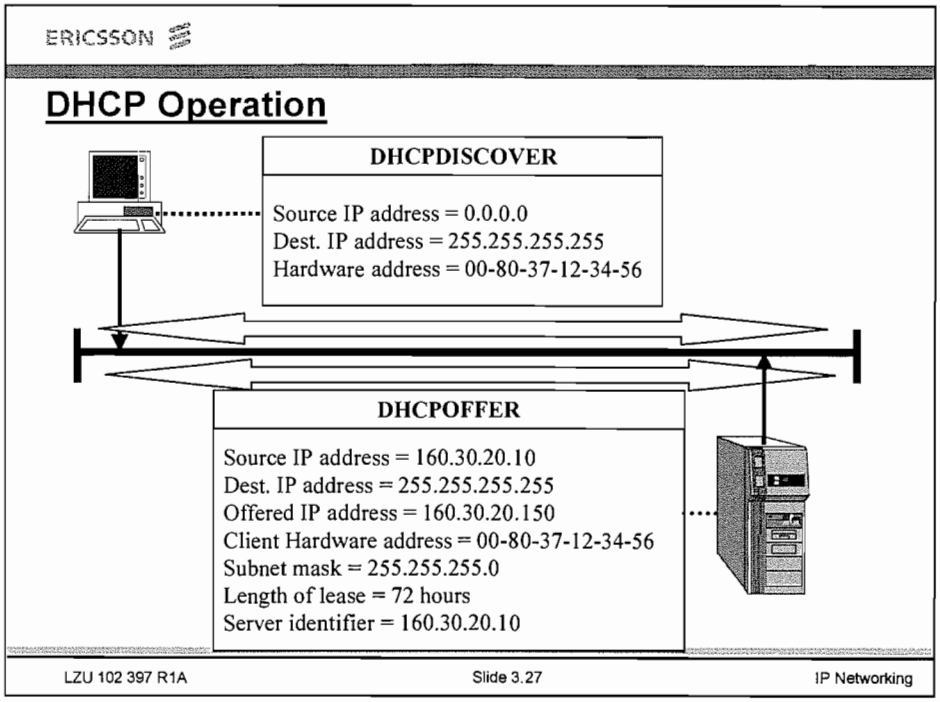


Figure 3-26.

Notes:

DHCP process (cont)

IP Lease Acknowledgement (Successful)

The DHCP server with the accepted offer broadcasts a successful acknowledgement to the client in the form of a DHCPACK message. This message contains a valid lease for an IP address and possibly other configuration information. When the DHCP client receives the acknowledgement, TCP/IP is completely initialised and is considered a bound DHCP client. Once bound, the client can use TCP/IP to communicate on the inter-network. The client stores the IP address, subnet mask and other IP addressing information locally.

IP Lease Acknowledgement (Unsuccessful)

An unsuccessful acknowledgement (DHCPNACK) is broadcast if:

- The client is trying to lease its previous IP address and the IP address is no longer available
- The IP address is invalid because the client has been physically moved to a different subnet.
- When the client receives an unsuccessful acknowledgement, it returns to the process of requesting an IP lease.

IP Lease Renewal

All DHCP clients attempt to renew their lease when 50 percent of the lease time has expired. To renew its lease, a DHCP client sends a DHCPREQUEST message directly to the DHCP server from which it obtained the lease. If the original DHCP server cannot renew a lease, the client still uses the address, as 50 percent of the lease life is still available.

The client will attempt to contact any available DHCP server when 87.5 percent of the lease time has expired (broadcast).

If this is unsuccessful and the lease time expires, the DHCP client can no longer use the IP address and communication over TCP/IP stops until a new IP address can be assigned to the client.

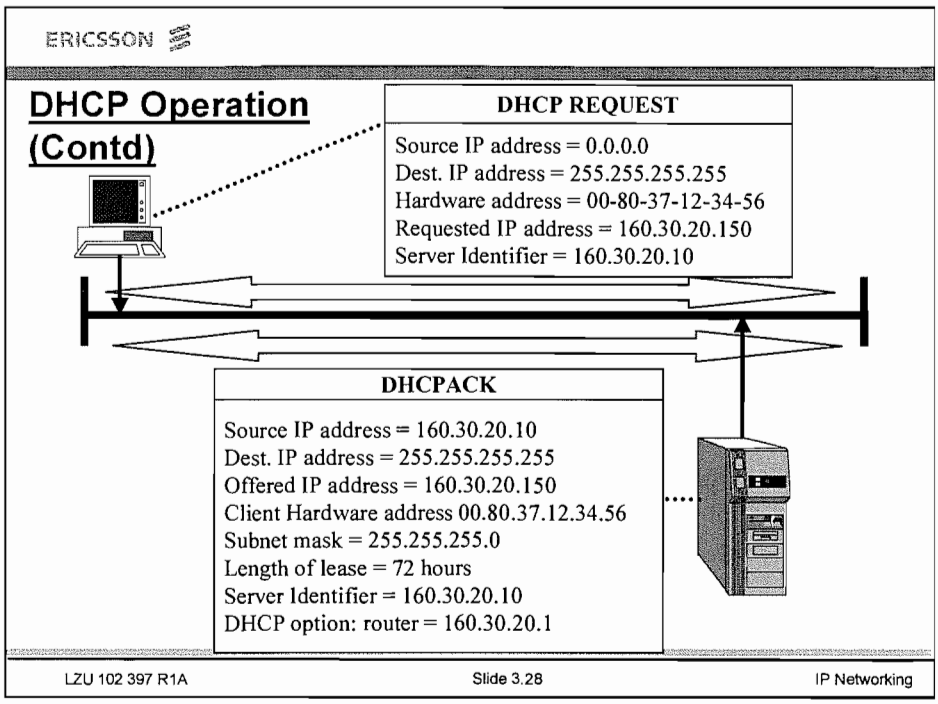


Figure 3-27.

Notes:

DHCP Interaction Through Routers

Routers can be configured to act as 'relay agents' in order to allow DHCP servers located on one IP network to serve configuration requests from remote networks.

A relay agent that conforms to RFC 1542 relays DHCP packets to a remote network even though they are broadcast packets. Before relaying a DHCP message from a DHCP client, the agent examines the gateway IP address field. If the field has an IP address of 0.0.0.0 the agent fills it with the router's IP address.

When the DHCP server receives the message it examines the relay IP address field in order to see if it has a DHCP scope (a pool of IP addresses) that can be used to supply an IP address lease. If the DHCP server has multiple scopes, the address in the relay IP address field identifies the DHCP scope from which to offer an IP address lease.

This process allows one DHCP server to manage different scopes for different networks.

When it receives the DHCP Discover message, the DHCP server sends a DHCP Offer directly to the relay agent identified in the gateway IP address field, and the agent relays the message to the client. The client's IP address is unknown, thus it has to be broadcast on the local subnet.

Similarly a DHCP request message is relayed from client to server and a DHCP ACK message is relayed from server to client according to RFC 1542.

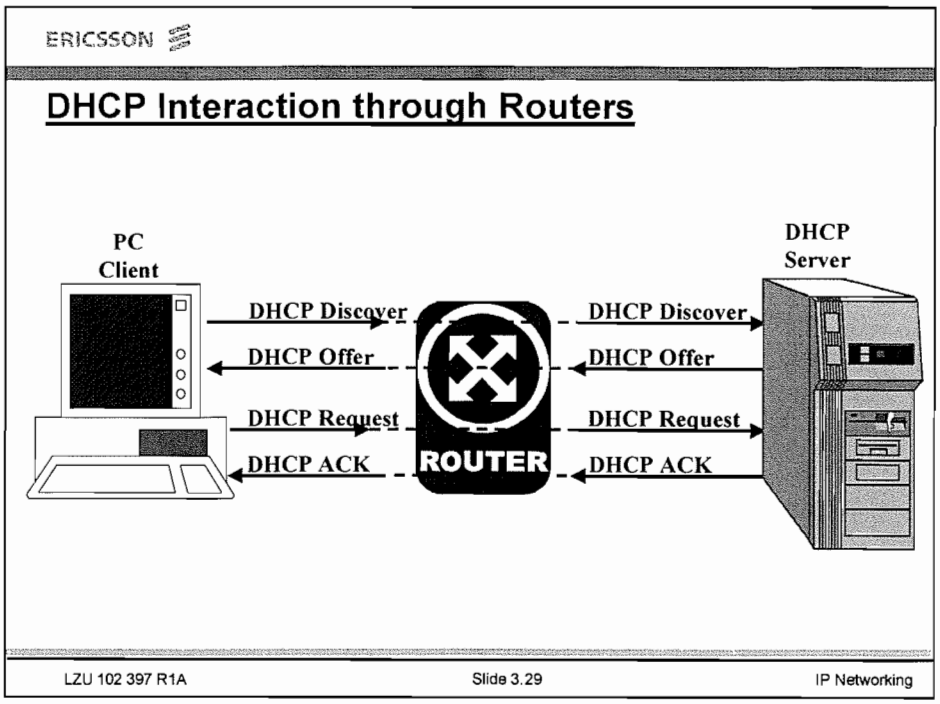


Figure 3-28.

Notes:

DHCP Message Format

Most of the fields (all except two) in a DHCP message are identical to fields in a BOOTP message.

DHCP interprets BOOTP's UNUSED field as a 16-bit FLAGS field. Only the high-order bit of the flags field has been assigned a meaning. A client can set this bit to request that the server respond using hardware broadcast instead of hardware unicast.

The DHCP OPTIONS field has the same format as the VENDOR SPECIFIC AREA in BOOTP, and DHCP honours all the vendor specific information items defined for BOOTP. As in BOOTP each option consists of a 1-octet code field and a 1-octet length field followed by octets of data which comprise the option. The option used to define a DHCP message type consists of exactly three octets. The first octet contains the code 53, the second contains the length 1, and the third contains a value (listed below) used to identify one of the possible DHCP messages.

1. DHCPDISCOVER
2. DHCPOFFER
3. DHCPREQUEST
4. DHCPDECLINE
5. DHCPACK
6. DHCPNACK
7. DHCPRELEASE

ERICSSON

DHCP Message Format

0	8	16	24	31
OP	HTYPE	HLEN	HOPS	
TRANSACTION ID				
SECONDS		FLAGS		
CLIENT IP ADDRESS				
YOUR IP ADDRESS				
SERVER IP ADDRESS				
ROUTER IP ADDRESS				
CLIENT HARDWARE ADDRESS (16 OCTETS)				
SERVER HOST NAME (64 OCTETS)				
BOOT FILE NAME (128 OCTETS)				
OPTIONS (VARIABLE)				

LZU 102 397 R1A Slide 3.30 IP Networking

Figure 3-29.

Notes:

DOMAIN NAME SYSTEM

IP Addresses and Symbolic Names

Each computer is assigned an Internet Protocol address, which appears in each IP packet sent to that computer. Anyone who has used the Internet knows that users do not need to remember or enter IP addresses.

Computers are also assigned symbolic names. Application software allows a user to enter one of the symbolic names when identifying a specific computer. Although symbolic names are convenient for humans, they are inconvenient for computers.

The underlying network protocols only understand addresses, so some mechanism to map symbolic names to IP addresses is required.

Domain Name Resolution

Application software translates symbolic computer names into equivalent Internet addresses. A database for implementing the naming scheme is distributed across the Internet.

This method of mapping the symbolic names to IP addresses through a distributed database is known as the Domain Name System (DNS).

Whenever an application program needs to translate a name, the application becomes a client of the naming system. The client sends a request message to a name server, which finds the corresponding address and sends a reply message. If a name server cannot answer a request, it temporarily becomes the client of another name server, until a server is found that can answer the request.

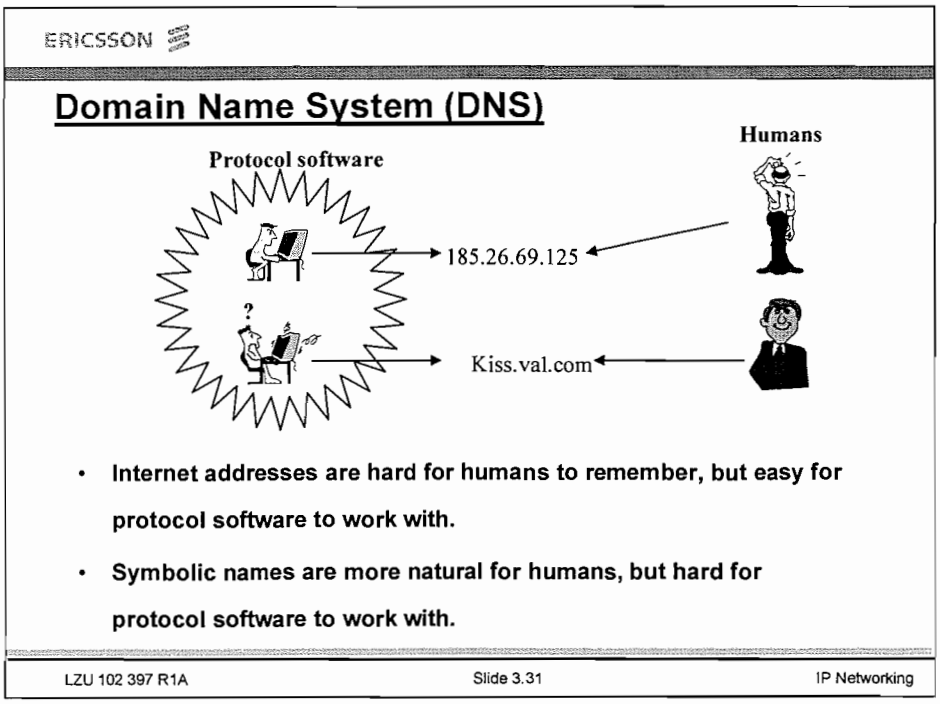


Figure 3-30.

Notes:

Domain Name System (cont)

As mentioned previously, the naming scheme used in the Internet is called the Domain Name System (DNS). The DNS is based on a hierarchical scheme, with the most significant part of the name on the right. The leftmost segment is the name of the individual computer. Other segments in a domain name identify the group that owns the name. For example, the Burger Department at Pizza has the domain name Burger.Krusty.cookie.Pizza.ie.

Basically, the Internet is divided into hundreds of top-level domains where each domain covers many hosts. Each domain is partitioned into subdomains, and these are further partitioned, and so on. There are two types of top-level domains: generic and country. The seven three-character generic domains are as follows:

- com (commercial organisation)
- edu (educational institution)
- gov (government organisation)
- mil (military group)
- net (major network support centre)
- org (organisation other than those above)
- int (international organisation)

Country domains consist of a two-letter entry for every country, as defined in ISO 3166. For example, .ie designates Ireland.

Each domain is named by the path up to the root. The path segments are separated by periods. For example, the Ericsson engineering department may have the domain name eng.ericsson.com.

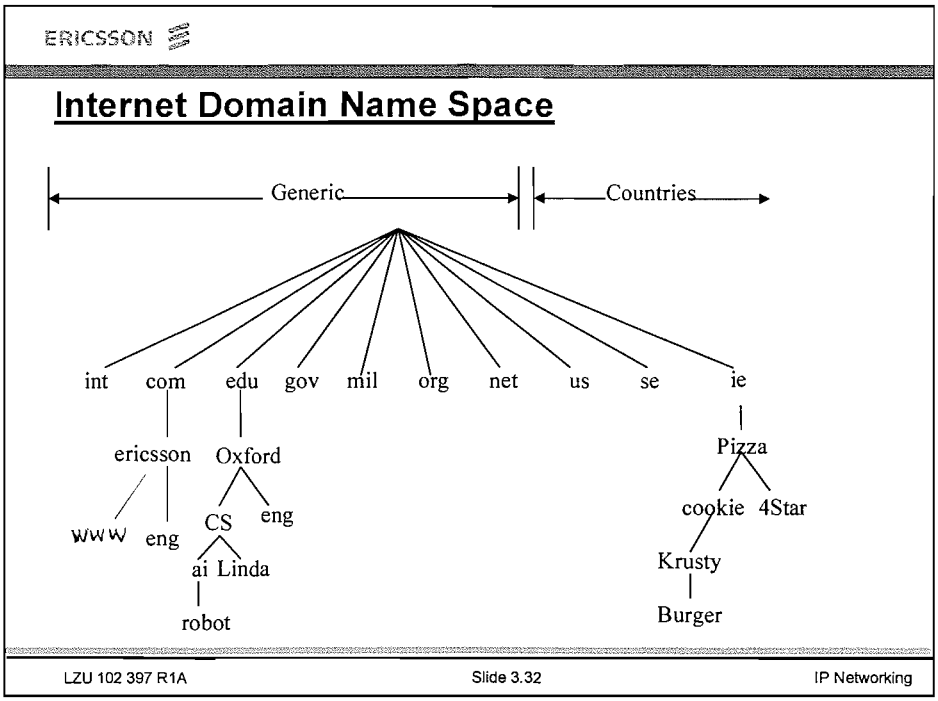


Figure 3-31.

Notes:

Resolving a Name

The translation of a domain name into an equivalent IP address is called name resolution. The name is said to be resolved to an address.

A host asking for DNS name resolution is called a resolver. Each resolver is configured with the address of a local domain name server. If a resolver wishes to become a client of the DNS server, the resolver places the specified name in a DNS request message and then sends the message to the local server.

The resolver then waits for the server to send a DNS reply message that contains the answer. DNS servers support both UDP and TCP.

TCP can break data into segments, and it can transfer any amount of data using multiple segments. Resolvers use UDP by default, which transmits 512 bytes in one packet. If the data exceeds 512 bytes, the server transmits the first 512 bytes and sets the truncated bit in the packet. The resolver then retransmits the request using TCP.

If the requested host name is contained by the name server's database, the server is said to be an authority for that host. When an incoming request specifies a name for which a server is an authority, the server answers the request directly. That is, the server looks up the name in its local database, and sends a reply to the resolver.

However, when a request arrives for a name outside the set for which the server is an authority, further client-server interactions result.

The two general approaches to dealing with this problem are 'recursive', in which the server pursues the query for the client at another server, and 'iterative', in which the server refers the client to another server and lets the client pursue the query. The DNS requires implementation of the iterative approach.

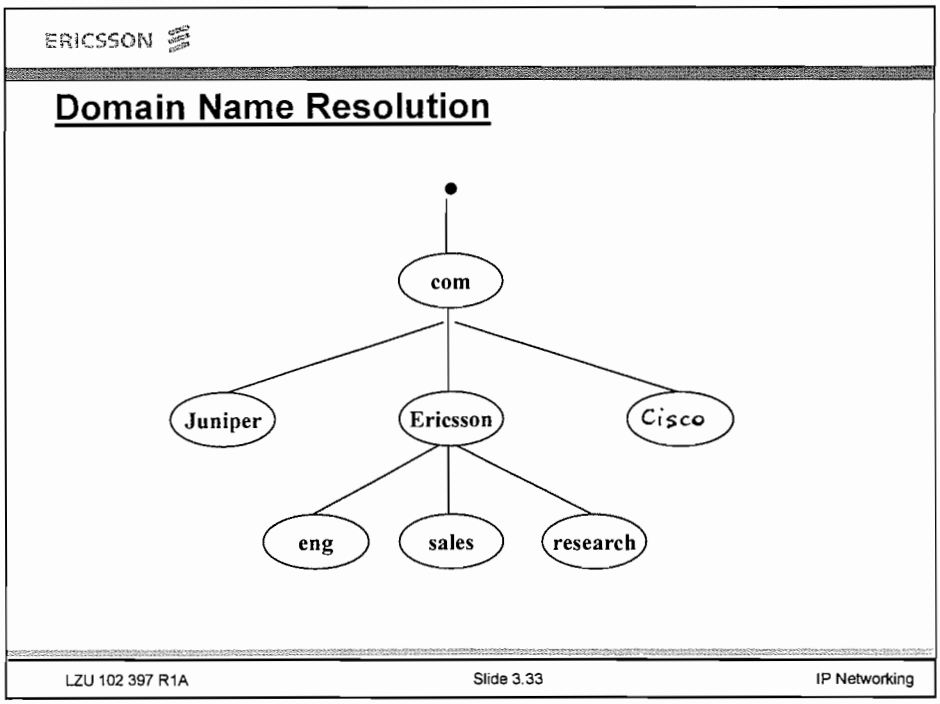


Figure 3-32.

Notes:

DNS Message Format

The identification field is set by the client and returned by the server. The identification field lets the client match responses to requests.

The flags field is divided into numerous sub-fields, which tell if the response is authoritative, if it is truncated or not, if recursion mode is available at the server, and if the query is desired to be recursive or iterative.

The next fields specify the number of entries that complete the record. For a DNS query, the number of questions is normally 1 and the number of answers zero. For a DNS reply, the number of answers is at least 1.

Each question has a query type and each response has a type. The responses the DNS server provides are called Resource Records (RRs).

ERICSSON

DNS Message Format

- There is one DNS message defined for both queries and responses.
- The message has a fixed 12-byte header followed by variable length fields.

identification	flags
number of questions	number of answers
number of authority RRs	number of additional RRs
question(s)	
answer(s)	
authority RRs	
additional RRs	

LZU 102 397 R1A Slide 3.34 IP Networking

Figure 3-33.

Notes:

Question and Answer Portion of a DNS Message

Each question has a query type and each response has an answer type. There are about 20 different values, some of which are now obsolete. The diagram opposite shows the most common query types. The most common query type is an A type, which means that an IP address is desired for the queried name.

The NS query type is made to find out the authoritative name server for a domain.

A common request from a secondary DNS to the primary is the AXFR (also called X-FER) type request. For this type of query, the secondary updates its database based on the response from the primary.

To support the storage of IPv6 addresses in DNSs, a new resource record type was defined to map a domain name to an IPv6 address. The AAAA or A4 resource record type is a new record specific to the Internet class that stores a single IPv6 address. The value of the type is 28 decimal. The returned address must be a 128-bit IPv6 format address.

ERICSSON

Question and Answer Portion of a DNS Message

- Each question has a query type and each response has a type.
- The answer(s) the DNS provides are called resource records.

Type	Value	Description
A	1	IP address
NS	2	Name server
AXFR	252	Request for zone transfer

- DNS servers which support IPv6 have a newly defined question / answer type - AAAA or A4 - to be able to ask and provide IPv6 addresses (an IPv6 address is four times longer than IPv4).

LZU 102 397 R1A Slide 3.35 IP Networking

Figure 3-34.

Notes:

DOMAIN NAME RESOLUTION

How does a DNS server know which other DNS server is the authority for a given name? The answer is that it does not. Each server, however, knows the address of a root server. Knowing the location of a root server is sufficient because the name can be resolved from there.

1 The resolver (DNS client) sends a recursive DNS query to its local DNS server asking for the IP address of 'server1.eng.ericsson.com'. The local name server is responsible for resolving the name and cannot refer the resolver to another name server.

2 The local name server is not an authority for the name so it sends an iterative query for server1.eng. ericsson.com to a root name server.

3 The root name server has authority for the root domain and replies with the IP address of a name server for the top-level domain.

4 The local name server sends an iterative query for 'server1.eng. ericsson.com' to the com name server.

5 The com name server replies with an IP address for the name server servicing the ericsson.com domain.

6 The local name server sends an iterative query for 'server1.eng. ericsson.com' to the ericsson.com name server.

7 The ericsson.com name server replies with an IP address for the name server servicing the eng. ericsson.com domain.

8 The local name server sends an iterative query for 'server1.eng. ericsson.com' to the eng.ericsson.com name server.

9 The eng.ericsson.com name server replies with the IP address of the requested host.

10 The local name server caches the IP address and sends it to the resolver.

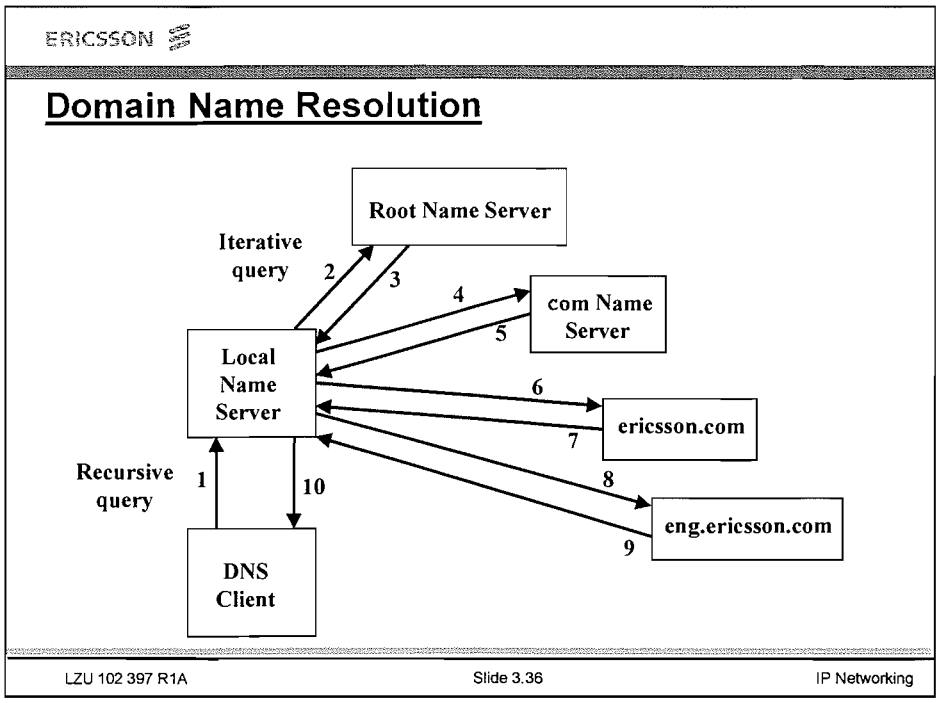


Figure 3-35.

Notes:

DNS CACHING

Internet name servers use name caching to reduce the traffic on the Internet and to improve performance. Each server maintains a cache of recently used names as well as a record of where the mapping information for that name was obtained.

When a client asks the server to resolve a name, the server first checks to see if it has authority for that name. If not, the server checks its cache to see if the name has been resolved recently.

Servers report cached information to clients, but mark it as a non-authoritative binding, and give the domain name of the server, S, from which they obtained the binding. The local server also sends along additional information that informs the client of the binding between S and an IP address. Therefore, clients receive answers quickly, but the information may be out of date.

If efficiency is important, the client chooses to accept the non-authoritative answer and proceed.

If accuracy is important, the client chooses to contact the authority and verify that the binding between name and address is still valid.

To keep the cache correct, servers time each entry and dispose of entries that exceed a reasonable time.

Servers allow the authority for an entry to configure this time-out. When an authority responds to a request, it includes a time to live (TTL) value in the response. The TTL specifies how long the authority guarantees that the binding will be valid.

Authorities can thus reduce network overhead by specifying long time-outs for entries that they expect to remain unchanged, and can improve correctness by specifying short time-outs for entries that they expect to change frequently.

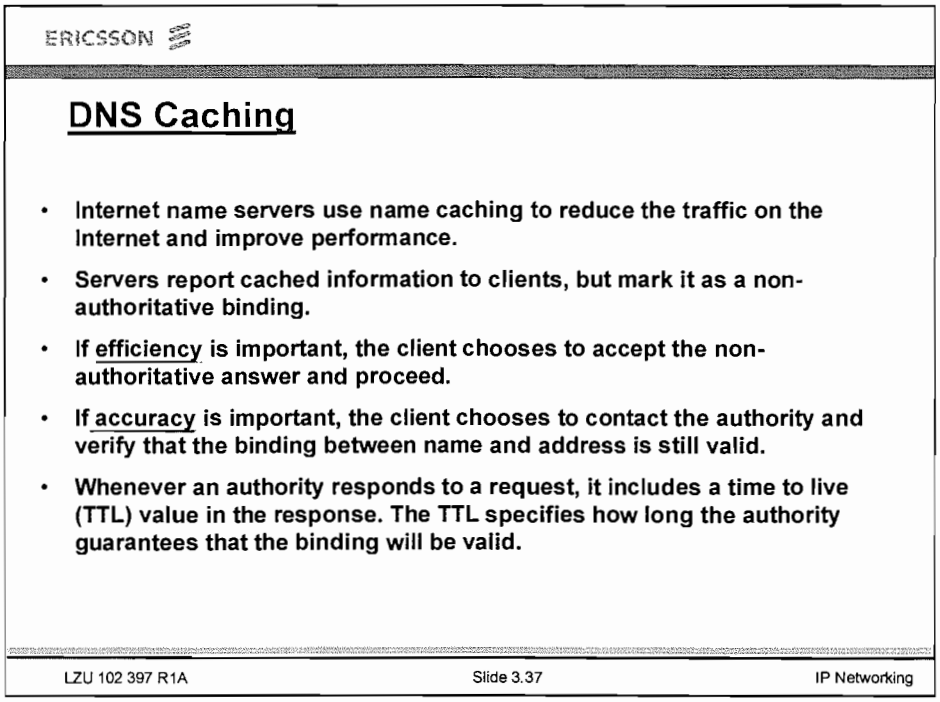


Figure 3-36.

Notes:

Zone Transfer Messages

A subtree of the DNS tree that is administered separately is called a zone. At least one name server has authority over a zone. The person responsible for a zone must provide a primary name server for that zone and one or more secondary name servers.

The primary and secondaries must be independent and redundant servers so that a single point of failure should not affect availability of name service for a zone.

When a new host is added to the zone, the administrator adds the appropriate information to the primary's database. This new information must be distributed to all secondary name servers. The secondary name servers query the primary on a regular, configurable time interval, and if the primary contains newer data, the secondaries obtain the new data using zone transfer request messages.

When the primary's database is updated, a field containing, for example, the date of the last update is also updated. The secondaries always check this field when sending a zone transfer request messages to the primary. If a change in the primary's database is identified, a copy of the database is downloaded. Because accuracy is essential, TCP must be used for this kind of data transfer.

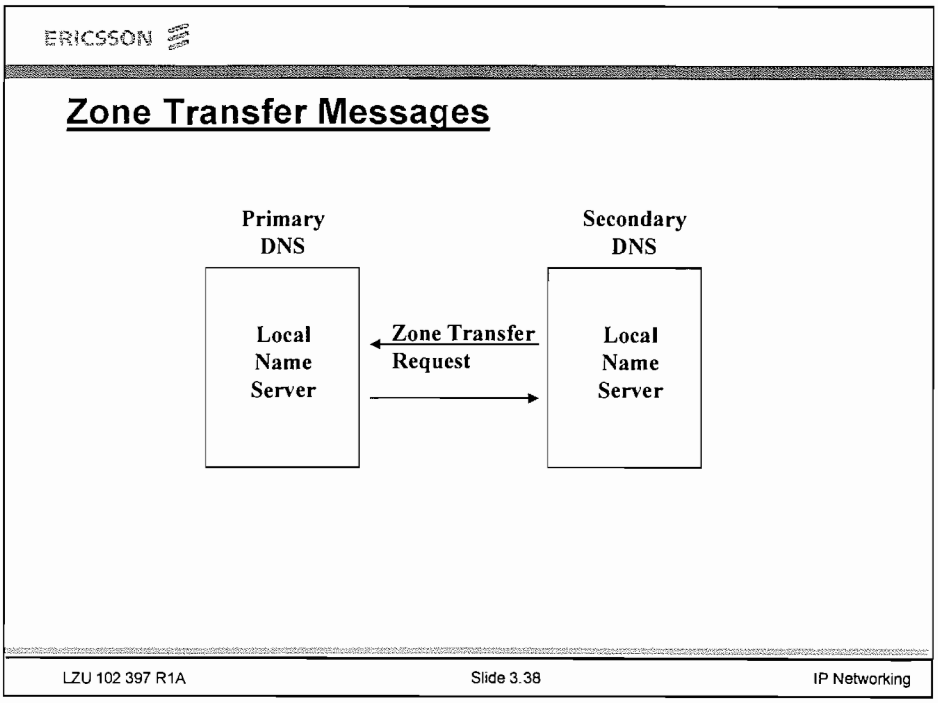


Figure 3-37.

Notes:

INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

The Internet Control Message Protocol (ICMP) reports errors and sends control messages on behalf of IP. ICMP does not attempt to make IP a reliable protocol. It simply attempts to report errors and provide feedback on specific conditions. ICMP messages are carried as IP packets and are therefore unreliable.

ICMP Message Format

Message Element	Bits	Purpose
Type	8	Specifies the type of ICMP message
Code	8	Specifies the condition inside one type of ICMP message.
Checksum	16	A checksum carried out on the ICMP header only.
Identifier	16	Used by the sender to match replies to requests.
Sequence number	16	Used by the sender to match replies to requests.
Optional Data	-	Specifies Contains information to be returned to the sender.

ERICSSON

Internet Control Message Protocol (ICMP)

- Reports errors and sends control messages on behalf of IP.
- ICMP messages are encapsulated within an IP packet.
- One of the most frequently-used debugging tools uses ICMP.
- ICMP message format:

IP Header.....		
Type	Code	Checksum
Identifier		Sequence Number
Optional Data		

LZU 102 397 R1A Slide 3.39 IP Networking

Figure 3-38.

Notes:

ICMP Message Types

Although each has its own format, all ICMP messages begin with the same three fields:

- An 8-bit integer message type field identifying the message
- An 8-bit code field providing further information about the message type
- A 16-bit checksum field

Destination Unreachable

When a router cannot forward or deliver an IP packet, it sends a destination unreachable ICMP message back to the original source.

Source Quench Message

A host or router uses source quench messages to report congestion to the original source and to request it to reduce its current rate of packet transmission.

Redirect Message

When a router detects a host using a non-optimal route, it sends the host an ICMP redirect message, requesting that the host change its routes. The router also forwards the original packet onto its destination.

Time Exceeded Message

Whenever a router discards a packet because its hop count has reached zero or because a timeout occurred while waiting for fragments of a packet, it sends an ICMP time exceeded message back to the packet's source.

Parameter Problem Message

When a router or host finds problems with a packet not covered by previous ICMP error messages (for example, an incorrect packet header), it sends a parameter problem message to the original source.

ERICSSON

ICMP Message Types

TYPE FIELD	ICMP Message Types
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect (change a route)
8	Echo Request
11	Time exceeded for a packet
12	Parameter problem on a packet

See http://www.iana.org/assignments/icmp-parameters
for full list of message types

LZU 102 397 R1A Slide 3.40 IP Networking

Figure 3-39.

Notes:

Echo Request and Reply Messages

One of the most frequently used debugging tools invokes the ICMP echo request and echo reply messages.

A host or router sends an ICMP echo request message to a specified destination.

Any device that receives an echo request formulates an echo reply and returns it to the original sender.

If the sender does not receive a reply it means that the destination is unreachable. The request also contains an optional data area.

The reply contains a copy of the data sent in the request. On many devices, the command that users invoke to send an ICMP echo request is named ping (packet Internet groper).

Sophisticated versions of ping send a series of ICMP echo requests, capture responses, and provide statistics about packet loss

ERICSSON

Echo Request and Reply Message Format

IP Header.....		
Type = 8 (or 0)	Code = 0	Checksum
Identifier		Sequence Number
Optional Data		

- ***These messages test whether a destination is reachable and responding, by sending ICMP echo requests (type 8) and receiving back ICMP echo replies (type 0).***
- **This test is carried out by using the Ping command.**

LZU 102 397 R1A Slide 3.41 IP Networking

Figure 3-40.

Notes:

Unreachable Messages

When a router cannot forward or deliver an IP packet, it sends a destination unreachable message back to the original source. The code field in the destination unreachable message contains an integer that further describes the problem, as shown in the diagram.

Network unreachable errors usually imply routing failures, while host unreachable errors imply delivery failures. Destinations may be unreachable for the following reasons:

- Hardware is temporarily out of service.
- The sender specified a non-existent destination address.
- The router does not have a route to the destination network.

Most of the messages are self-explanatory. For example, if the packet contains a source route option (list of routers, which the packet must pass through) with an incorrect route then it may trigger a source route failed message.

If a router needs to fragment a packet but the 'don't fragment' (DF) bit is set, then the router sends a fragmentation needed message back to the source.

ERICSSON

Reports of Unreachable Destinations

Code Value	Meaning
0	Network unreachable
1	Host unreachable
2	Protocol unreachable
3	Port unreachable
4	Fragmentation needed and DF set
5	Source route failed
6	Destination network unknown
7	Destination host unknown
8	Source host isolated
9	Communication with destination network administratively prohibited
10	Communication with destination host administratively prohibited
11	Network unreachable for type of service
12	Host unreachable for type of service

LZU 102 397 R1A Slide 3.42 IP Networking

Figure 3-41.

Notes:

TRACEROUTE

Traceroute is an application that uses ICMP and the TTL field in the IP header in order to make visible a possible path that IP packets follow from one host to another.

When a router gets an IP packet whose TTL is either 0 or 1 the router must not forward the packet. Instead the router discards the packet and sends an ICMP 'time exceeded' message back to the originating host.

The key to traceroute is that the IP packet containing this ICMP message has the router's IP address as the source address.

Traceroute operates as follows.

It sends an IP packet with a TTL of 1 to the destination host. The first router to handle the packet decrements the TTL, discards the packet, and sends back an ICMP time exceeded. This identifies the first router in the path.

Traceroute then sends a packet with a TTL of 2, and the IP address of the second router is found. This continues until the packet reaches the destination host.

Even though the arriving IP packet has a TTL of 1, the destination host will not discard the packet and generate the ICMP time exceeded, since the packet has reached its final destination.

Instead traceroute chooses a destination UDP port number with an unlikely value (larger than 30,000), making it improbable that an application at the destination is using that port. This causes the destination host's UDP module to generate an ICMP 'port unreachable' message when the packet arrives.

All traceroute needs to do is to differentiate between the received ICMP messages (time exceeded versus port unreachable) to know that the packet has reached its final destination.

Some implementations of traceroute do not use UDP with unlikely port number for the destination device. Instead they transmit an ICMP echo request with incrementing TTLs and when this eventually reaches its destination, an echo reply is received by the originator. This method also shows a possible path to the destination.

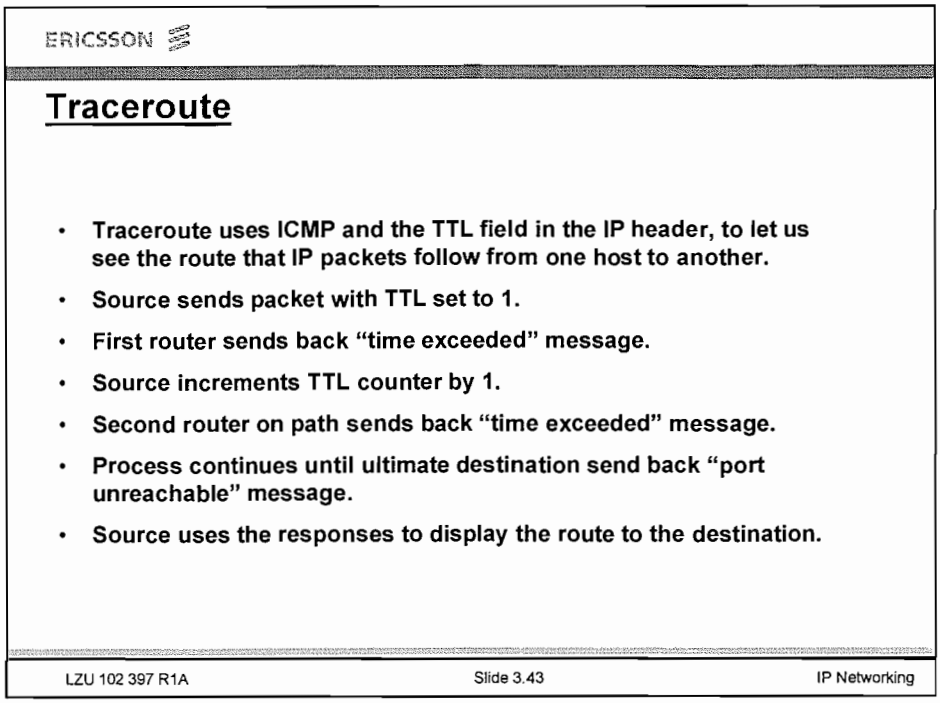


Figure 3-42.

Notes:

Intentionally Blank

4 Applications

After completing this chapter you will be able to:

- Understand remote command execution (Telnet and rlogin)
- Understand remote File Transfer Protocol (FTP, TFTP)
- Understand protocols for mail transfer (SMTP, POP3, IMAP4)
- Understand HTTP protocols

Intentionally Blank

TELNET

The TCP/IP protocol suite includes a simple remote terminal protocol called Telnet. Telnet allows a user at one site to establish a TCP connection to a login server at another. Telnet then passes the keystrokes from the user's keyboard directly to the remote computer, as if they had been typed on a keyboard attached to the remote computer. Telnet also carries output from the remote computer back to the user's screen.

Telnet client software allows the user to specify the remote computer, either by giving its domain name or IP address. In the login procedure, the username and password are transferred unsecured (plain text) through the network.

Telnet offers a number of basic services. It defines a network virtual terminal (NVT) that provides a standard interface to remote systems.

Client programs do not have to understand the details of all possible remote systems, as they are built to use the standard interface. Telnet includes a mechanism that allows the client and server to negotiate options.

Telnet provides a set of standard options. For example, one of the options controls whether data passed across the connection uses the standard 7-bit ASCII character set or an 8-bit character set. Also, Telnet treats both ends of the connection symmetrically. In particular, Telnet does not force client input to come from a keyboard, nor does it force the client to display output on a screen. Thus Telnet allows an arbitrary program to become a client. Furthermore, either end can negotiate options.

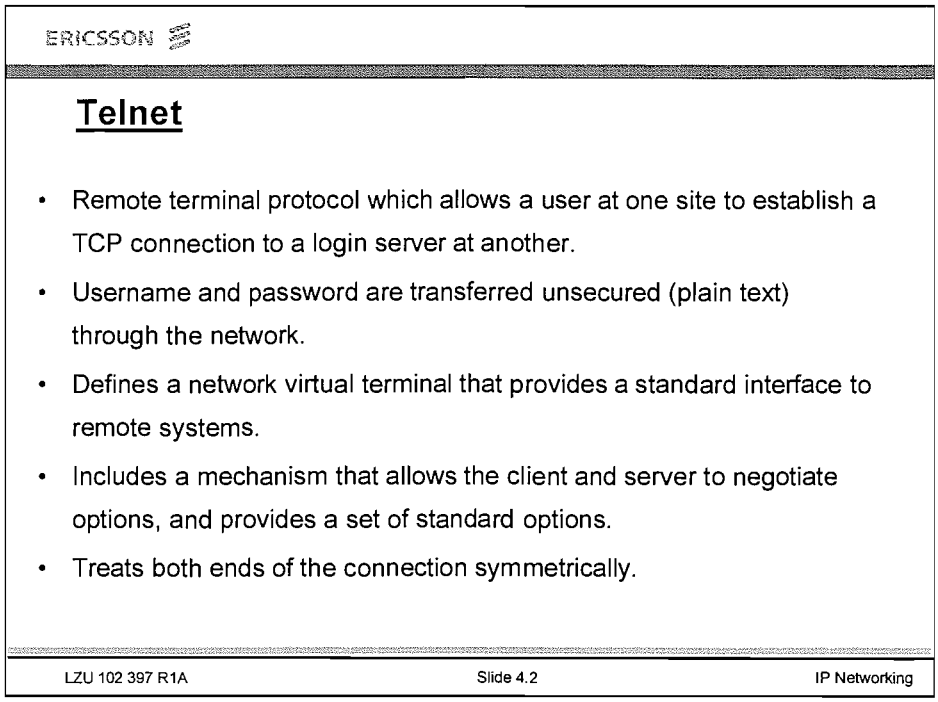


Figure 4-1.

Notes:

Telnet Operation

The Telnet protocol provides a standardised interface through which a program on one host (the Telnet client) accesses the resources of another host (the Telnet server), as though the client were a local terminal connected to the server. Telnet is also used for logging into bridges, routers and other network devices for management purposes such as configuration. For example, a user on a workstation on a LAN can connect to a host attached to the LAN as though the workstation were a terminal attached directly to the host. Telnet can be used across WANs as well as LANs.

Telnet works as follows:

1. A user accesses the Telnet application and selects the destination host (IP address or computer name)

2. During the login session, the user interacts with the local Telnet service. The local Telnet sets up a TCP connection to the Telnet server (on port 23) at the remote host.

3. The local Telnet exchanges data with the remote Telnet service.

4. The remote Telnet service interacts with applications at the destination host. It is up to the local (client) and remote (server) Telnets to create the appropriate terminal emulation that enables the local terminal to talk to the remote application.

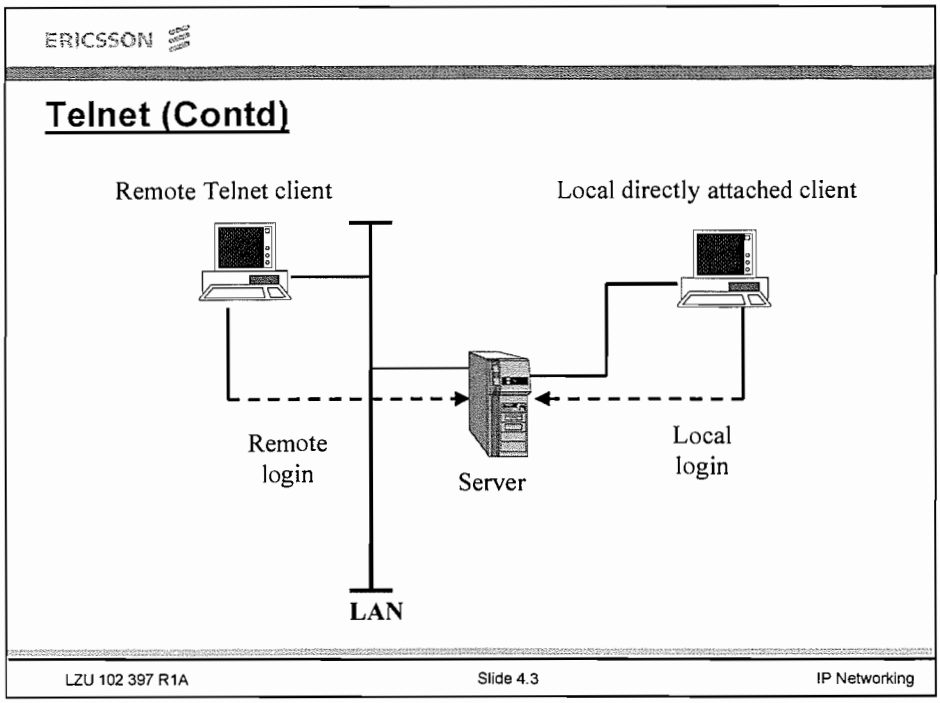


Figure 4-2.

Notes:

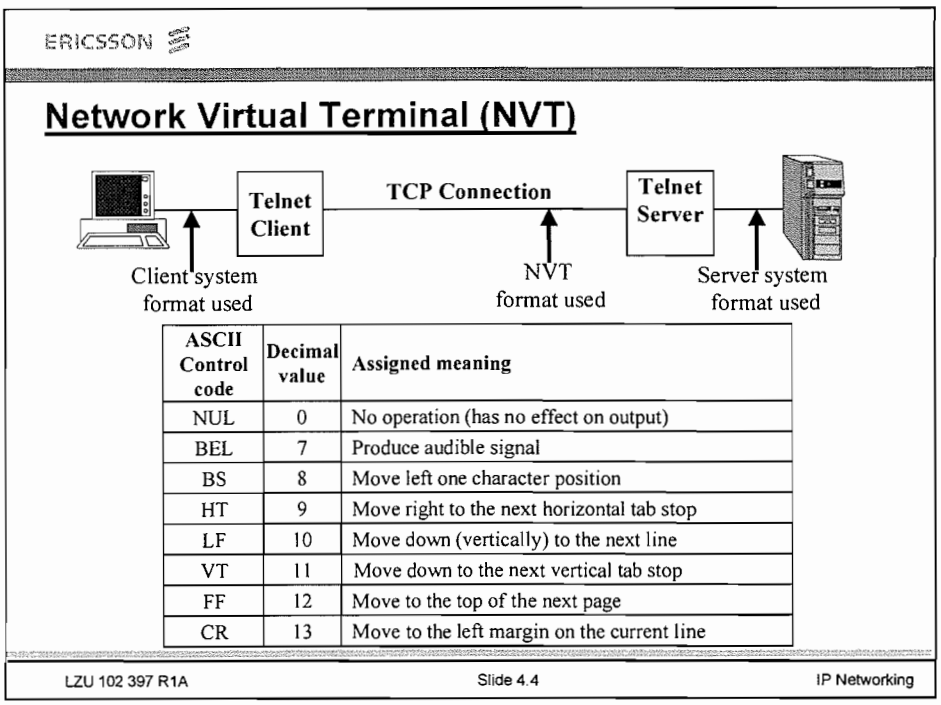
Network Virtual Terminal (NVT)

The characteristics of a locally attached terminal are known to the application, but the characteristics of a remote terminal are likely to be unknown to the application. Hence, when Telnet is used, the client converts the terminal characteristics of the user to those of a universal terminal type, called virtual terminal. The server converts the virtual terminal characteristics to make them appear as though generated by a local terminal. This feature is called network virtual terminal (NVT).

The NVT represents the lowest common denominator of existing terminal features:

- A bi-directional character device with a printer and a keyboard
- Operates in scroll mode
- Unlimited line/page length
- Uses seven-bit ASCII characters encoded in eight-bit octets

Since ASCII control characters vary between systems, Telnet has a precise definition of NVT control characters, which are shown in the diagram.



ASCII Control code	Decimal value	Assigned meaning
NUL	0	No operation (has no effect on output)
BEL	7	Produce audible signal
BS	8	Move left one character position
HT	9	Move right to the next horizontal tab stop
LF	10	Move down (vertically) to the next line
VT	11	Move down to the next vertical tab stop
FF	12	Move to the top of the next page
CR	13	Move to the left margin on the current line

Figure 4-3.

Notes:

Telnet Command Codes

During a connection either the user or the application can, at any stage, negotiate enhanced characteristics beyond those offered by the NVT. This is accomplished by embedded commands in the data stream. Telnet command codes are one or more octets in length, and are preceded by an Interpret As Command (IAC) character, which is an octet with each bit set equal to one (FF hex). If an octet equal to FF hex occurs in real data, an IAC character must precede it in order to prevent it being mistaken as an IAC character. The receiving side discards the added IAC character, and data integrity is preserved. The IAC character can be followed by a single command, or by a command and negotiable options. The diagram illustrates the Telnet command codes.

As the diagram shows, the signals generated by conceptual keys on an NVT keyboard each have a corresponding command. For example, to request that the server interrupt the executing program, the client must send the 2-octet sequence IAC IP (255 followed by 244). Additional commands allow the client and server to negotiate which options they use and to synchronise communication.

When a client negotiates what it wants for itself, the verb 'will' is used for enabling (response 'do' for accept and 'don't' for reject), and the verb 'won't' is used for disabling (response 'don't' for accept - it cannot be rejected).

When a client negotiates what it wants for the server, the verb 'do' is used for enabling (response 'will' for accept and 'won't' for reject), and the verb 'don't' is used for disabling (response won't is affirmative because the basic NVT must be supported).

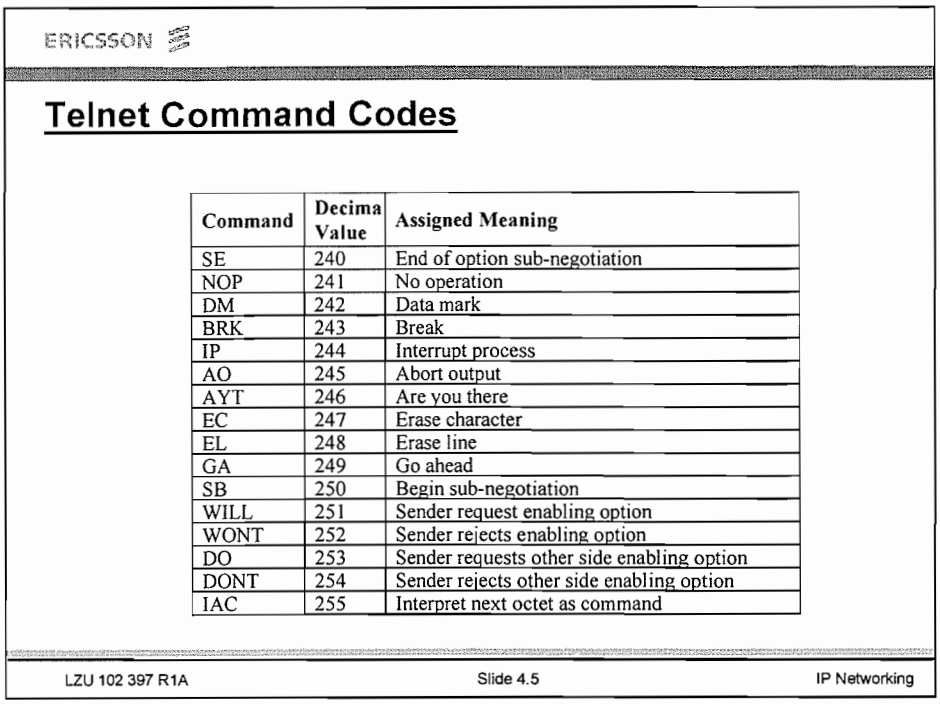


Figure 4-4.

Notes:

Telnet Options

In Telnet, options are negotiable, making it possible for the client and server to reconfigure their connection. The range of Telnet options is wide. Some options extend the Telnet capabilities in major ways, while others deal with minor details.

For example, the original protocol was designed for a half-duplex environment where it was necessary to tell the other end to 'go ahead' before it would send more data. One of the options controls whether Telnet operates in half-duplex or full-duplex mode.

Another option allows the server on a remote machine to determine the user's terminal type. The terminal type is important for software that generates cursor-positioning sequences, for example a full screen editor executing on a remote machine.

The diagram lists several of the most commonly implemented Telnet options. Each negotiable option has a code number, which immediately follows the command for option negotiation, that is IAC, <command>, <option code>. The negotiation process involves command and responses, because commands can be accepted or rejected. There are only four commands for option negotiation:

- Will
- Won't
- Do
- Don't

If one side tries to negotiate an option that the other does not understand, the side receiving the request can simply decline. Thus, it is possible to inter-operate newer, more sophisticated versions of Telnet clients and servers (that is software that understands more options) with older, less sophisticated versions. This works because all Telnet software understands a basic NVT protocol, therefore clients and servers can inter-operate even if one understands options that the other does not.

ERICSSON

Telnet Options

Name	Code	RFC	Assigned meaning
Transmit Binary	0	856	Change transmission to 8-bit binary
Echo	1	857	Allow one side to echo data it receives
Suppress-GA	3	858	Suppress go-ahead signal after data
Status	5	859	Request for status of a Telnet option from remote site
Timing-mark	6	860	Request timing mark to be inserted in return stream
Terminal-type	24	884	Exchange info.. about the terminal type being used
End-of-record	25	885	Terminate data sent with EOR code
Linemode	34	1116	Send complete lines instead of individual characters

LZU 102 397 R1A Slide 4.6 IP Networking

Figure 4-5.

Notes:

RLOGIN

An implementation of Telnet under UNIX called rlogin provides greater flexibility and permits direct access to the remote application's command interpreter. The UNIX server daemon is called telnetd, and the command to use telnetd is 'rsh'.

Rsh invokes a command interpreter on the remote UNIX machine and passes the command line arguments to the command interpreter, skipping the login step completely.

The format of a command invocation using rsh is:

rsh <machine> <command>

Thus typing; 'rsh unixserver1 ps' on any of the Unix machines on the same network executes the ps command on the machine unixserver1, with UNIX's standard input and standard output connected across the network to the user's keyboard and display. The user sees the output as if he or she was logged into the machine unixserver1.

Because protocols like rlogin understand both the local and remote computing environments, they communicate better than general purpose remote login protocols like Telnet. For example, rlogin understands the UNIX notions of standard input, standard output, and standard error, and uses TCP to connect them to the remote machine.

Thus it is possible to type, 'rsh unixserver1 ps > filename' and have output from the remote command directed into file 'filename'.

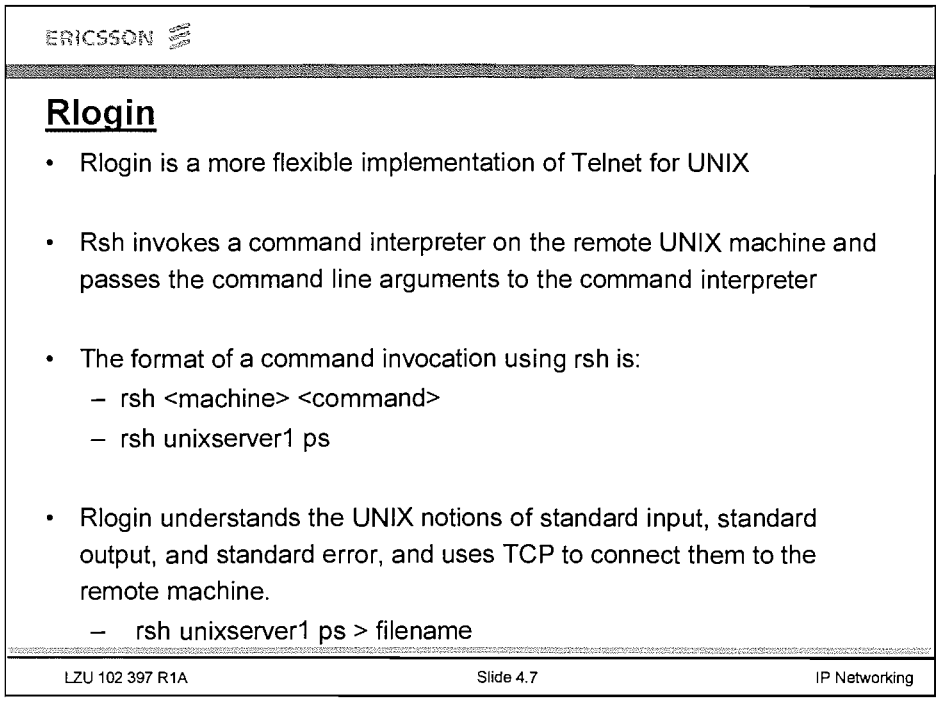


Figure 4-6.

Notes:

FILE TRANSFER PROTOCOL (FTP)

FTP is the Internet standard for file transfer. FTP is used to copy a complete file from one system to another system. FTP also offers other facilities beyond the transfer function itself.

Although FTP was designed to be used by programs, most implementations provide an interactive interface that allows humans to easily interact with remote servers.

FTP requires clients to authenticate themselves by sending a login name and password to the server before requesting file transfers. The user name and password is associated with a specific level of permissions on the server. The server refuses access to clients that cannot supply a valid login and password. Anonymous FTP is a means by which archive sites allow general access to their archives of information. These sites create a special account called "anonymous". User "anonymous" has limited access rights to the archive host, as well as some operating restrictions.

The username and password are transferred unsecured (plain text) through the network.

As well as file transfer, FTP is designed to allow a user to change directory, create, delete move and rename directories and files on the server. For example, a user can ask for a listing of all files in a directory on a remote computer. Also, the client usually responds to the input “help” by showing the user information about possible commands that can be used.

Format (Representation) Specification FTP allows the client to specify the type and format of the stored data. For example, the user can specify whether a file contains text or binary integers, and whether text files use the ASCII or EBCDIC character sets.

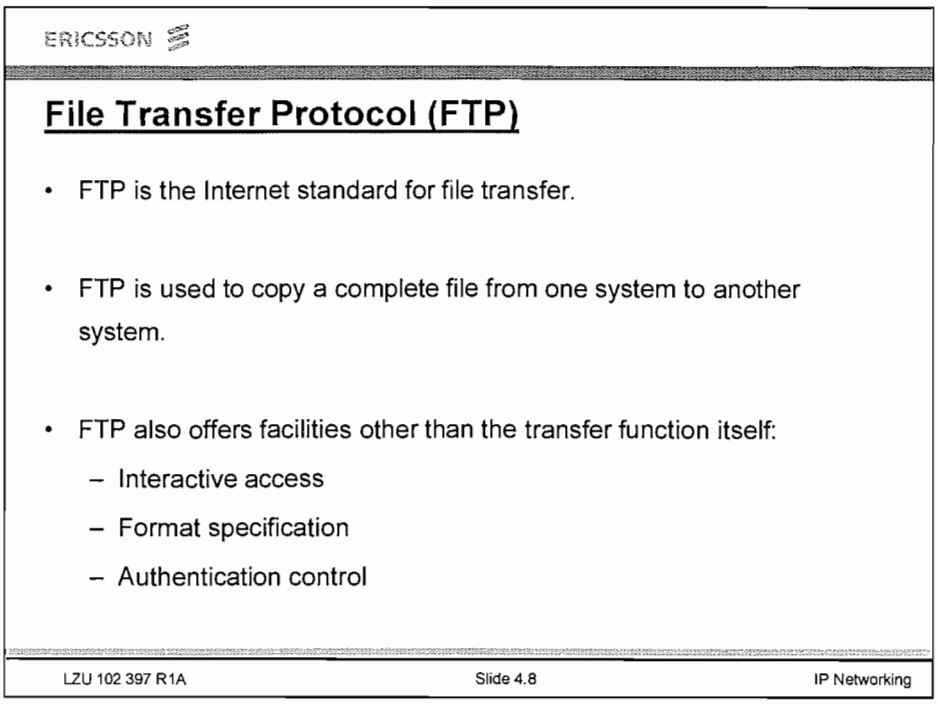


Figure 4-7.

Notes:

File Transfer Protocol (Contd)

The client control process connects to the server control process using one TCP connection, while the associated data transfer processes use their own TCP connection.

When a client forms an initial connection to a server the client uses a random, locally assigned, protocol port number, but contacts the server at a well-known port (21).

When the control process creates a new TCP connection for a given data transfer, it must assign protocol port numbers. The client obtains an unused port on its machine, and uses the port to contact the data transfer process on the server's machine. The data transfer process on the server machine can use the well-known port reserved for FTP data transfer (20).

In addition to passing user commands to the server, FTP uses the control connection to allow client and server control processes to co-ordinate their use of dynamically assigned TCP protocol ports and the creation of data transfer processes that use these ports.

FTP uses the Telnet Network Virtual Terminal (NVT) protocol format for passing data across the control connection. FTP does not allow option negotiation. It uses only the basic NVT definition.

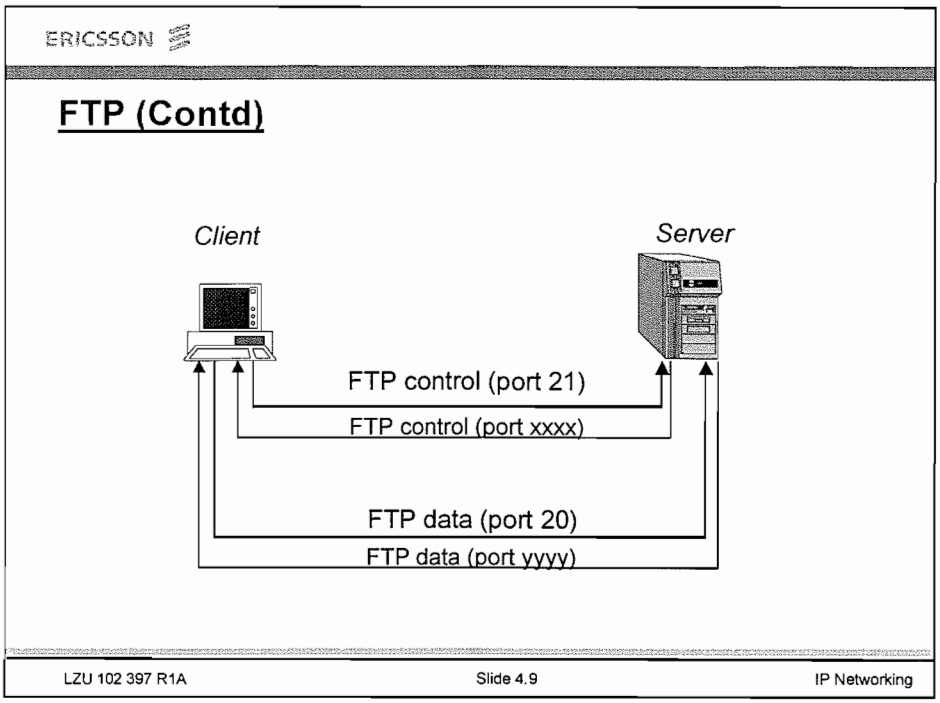


Figure 4-8.

Notes:

FTP Commands

FTP commands are three or four bytes of ASCII characters, some with optional arguments. Some of the most commonly used commands are listed below.

open	connect to remote FTP
disconnect	terminate FTP session
user	send new user information when already connected
cd	change remote working directory
lcd	change local working directory
pwd	print working directory on remote machine
prompt	force interactive prompting on multiple commands
get / mget	receive file / get multiple files
put / mput	send file / send multiple files
binary	set binary transfer type
ascii	set ascii transfer type
dir / ls	list contents of remote directory
help	get help
delete	delete a file on the remote directory
close	terminate ftp session
bye	terminate ftp session and exit

Note that the remote system may be case sensitive on the names of directories and files.

Most clients allow a user to issue commands to their local host by prefixing the command with exclamation symbol !

ERICSSON

FTP Commands

Command	Description
open	connect to remote FTP
disconnect	terminate FTP session
user	send new user info when already connected
cd	change remote working directory
lcd	change local working directory
pwd	print working directory on remote machine
get/mget	receive file/get multiple files
put/mput	send file/send multiple files
binary	set binary transfer type
ascii	set ascii transfer type
dir/ls	list contents of remote directory
help	get help
delete	delete a file on the remote directory
bye	terminate ftp session and exit

LZU 102 397 R1A Slide 4.10 IP Networking

Figure 4-9.

Notes:

FTP Replies

FTP replies are 3-digit numbers in ASCII, with an optional message following the number. The intent is that the software needs to look at the number to determine how to process the reply, and the optional string is for human consumption.

Since the clients normally output both the numeric reply and the message string, an interactive user can determine what the reply says by just reading the string (and not have to memorise what all the numeric reply codes mean).

Each of the three digits in the reply code has a different meaning.

The diagram shows the meanings of the first and second digits of the reply code. The third digit gives additional meaning to the error message. For example here are some typical replies, including a possible message string:

125 Data connection already open; transfer starting

200 Command OK

214 Help message (for human user)

331 Username OK, password required

425 Cannot open data connection

452 Error writing file

500 Syntax error (unrecognised command)

501 Syntax error (invalid arguments)

502 Unimplemented mode type

ERICSSON

FTP Replies

Reply	Description
1yz	Positive preliminary reply. The action is being started, but expect another reply, before sending another command.
2yz	Positive completion reply. A new command can be sent.
3yz	Positive intermediate reply. The command has been accepted but another command must be sent.
4yz	Transient negative completion reply.The requested action did not take place, but the error condition is temporary so the command can be reissued later.
5yz	Permanent negative completion reply. The command was not accepted and should not be retried.
x0z	Syntax errors
x1z	Information
x2z	Connections; Replies referring to the control or data connections.
x3z	Authentication and accounting. Replies for the login or accounting commands.
x4z	Unspecified
x5z	Filesystem status

LZU 102 397 R1A Slide 4.11 IP Networking

Figure 4-10.

Notes:

FTP Example

In the example in the diagram, FTP is used to download a copy of an RFC (Request for Comment) from the Internet. The user (local) commands are illustrated in bold font. From a DOS prompt, the command 'ftp rs.internic.net' is entered. The FTP client makes a TCP connection with the server. Normally a server would require identification in the form of 'user@host', but this server accepts the username 'anonymous' for general public access. All RFCs are under the directory RFC, so it is necessary to change to that directory with the command, 'cd rfc'.

The user command to transfer a file containing text for an RFC is 'get rfcnnnn.text', for example 'get rfc1878.text'.

Once FTP is started, the FTP client responds to the user as in an interactive system. Each command entered is followed by a response beginning with a three digit number.

Note that, for users with a web browser and an FTP client, it is a point and click intuitive process to FTP an RFC.

Click on [File] [Open Page], then type in ftp://ftp.rs.internic.net.

Anonymous Login

While the access authentication facilities in FTP make it more secure, strict enforcement prohibits an arbitrary client from accessing any file until that client obtains a login and password for the computer on which the server operates.

To provide access to public files, many TCP/IP sites allow anonymous FTP. Anonymous FTP access means a client does not need an account or password. Instead the user specifies login name "anonymous" and password as "guest" or their email address.

The server allows anonymous logins, but restricts access to only publicly available files. In many UNIX systems, the server restricts anonymous FTP by changing the file system root to a small, restricted directory, for example, /usr/ftp.

ERICSSON

FTP Example

```
> ftp rs.internic.net
Connected to rs.internic.net.
220-*****Welcome to the InterNIC Registration Host *****
*****Login with username "anonymous"
*****You may change directories to the following:
policy          - Registration Policies
templates       - Registration Templates
netinfo         - NIC Information Files
domain          - Root Domain Zone Files
220 And more!
User (rs.internic.net:(none)): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:xxxxxxxxxxxx
230 Guest login ok, access restrictions apply.
ftp> cd rfc
250 CWD command successful.
ftp> get rfc1878.txt
200 PORT command successful.
150 Opening ASCII mode data connection for rfc1878.txt (19414 bytes).
226 Transfer complete.
ftp: 19865 bytes received in 85.02Seconds 0.23Kbytes/sec.
```

LZU 102 397 R1A | Slide 4.12 | IP Networking

Figure 4-11.

Notes:

FTP Example (Contd)

The diagram illustrates the primary sequence of commands and responses, which take place between the client and server in order to obtain the RFC. The user commands for this FTP were illustrated in the last diagram. 'ftp ns.internic.net' has been entered by the user. The first line in the diagram is a login with username request from the server (USER). User commands are illustrated in lower case and client commands are in upper case.

All commands and responses prior to the retrieve command (RETR) use the FTP control connection.

The actual data transfer uses the FTP data connection. RETR by the client was preceded by a 'get' command from the user.

Upon completion of the data transfer, the data connection is closed.

The QUIT command from the client was preceded by a 'close' command from the user, which resulted in the response message, '221 Goodbye' from the client to the user (not shown). The user enters the command 'quit' (or 'bye') to disconnect from the server.

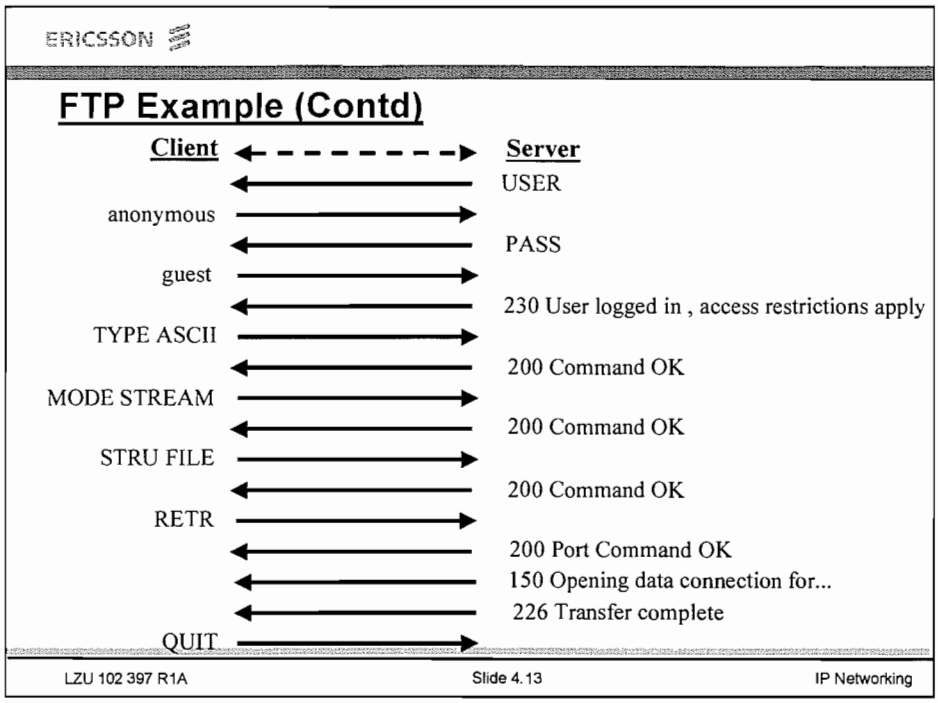


Figure 4-12.

Notes:

TRIVIAL FILE TRANSFER PROTOCOL (TFTP)

Trivial file Transfer Protocol (TFTP) is an extremely simple protocol to transfer files. TFTP differs from FTP in several ways:

- Communication between a TFTP client and server uses UDP (port 69) and not TCP. Therefore, TFTP lacks the FTP/TCP features such as sequencing, windowing and full duplex operation.
- TFTP only supports file transfer. That is, TFTP does not support interaction and does not have a large set of commands.
- TFTP does not have user authentication. A client does not send a login name or password. Therefore, a file can be transferred only if the file permissions allow global access.

Although TFTP is not as powerful as FTP, it was designed with UDP to make it simple and small. Implementations of TFTP (and its required UDP, IP and device driver) can fit in read-only memory. This makes TFTP suitable for bootstrapping diskless systems, for example, X-terminals.

The UDP port 69 is only used as the destination port in the initial request to the server. If the server grants the PUT or GET, request, it responds to the client with a randomly selected port greater than 1024 as its source port. The client now uses this new server port as the destination port during the transfer.

UDP checksum is calculated on the source and destination ports and length field of the UDP header and the IP address and protocol number from the IP header. Neither UDP nor TFTP calculate a checksum on the data in a TFTP transfer.

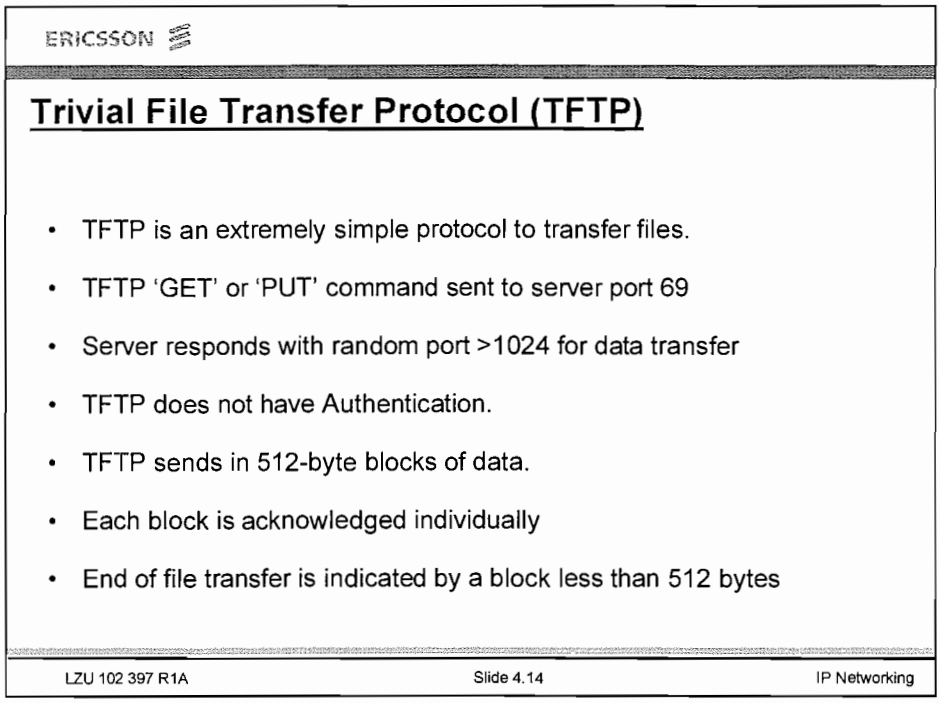


Figure 4-13.

Notes:

TFTP Process

With a total of only five commands, the rules of TFTP are simple. An example of a client performing a write to the TFTP server is illustrated in the diagram.

Note that both ends involved in a transfer are considered senders and receivers. One end sends data and receives acknowledgements, the other end receives data and sends acknowledgements.

The commands and their opcode numbers are:

1 Read

2 Write

3 Send

4 Acknowledge

5 Error

The TFTP process begins with the user sending either a read request (RRQ) or a write request (WRQ).

The RRQ is used to obtain a file from the server and the WRQ is used to send a file to the server. The example uses WRQ, code 2. The communication is synchronous in that every command sent must be acknowledged before the next command can be sent.

The WRQ request is acknowledged with an ACK response from the server that contains a block number equal to zero. An acknowledgement with a block number of zero signals to the client that the connection is initialised and the server is ready to receive.

Data is transferred in blocks of 512 octets. Each data block is numbered sequentially from one upwards. The receiver acknowledges each block with an ACK.

This process of sending an acknowledgement continues until a block with less than 512 octets is received, which is the signal that this is the last block of the message. If the data to be transferred is exactly divisible by 512, the sender transmits the last data block with a data size of zero (less than 512) to indicate the end of data transfer.

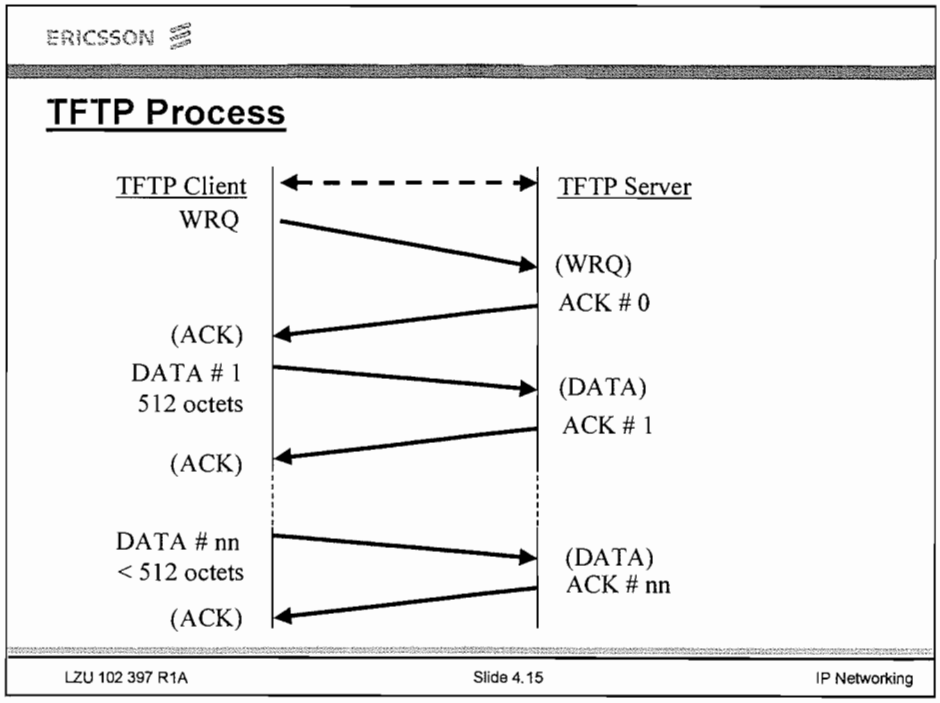


Figure 4-14.

Notes:

TFTP Message Format

The first two bytes of the TFTP message are an opcode. For an RRQ and a WRQ the filename specifies the file on the server that the client wants to read from or write to. This filename is terminated by an octet of zero.

The mode is one of the ASCII strings netascii or octet, again terminated by an octet of 0. Netascii means the data are lines of ASCII text with each line terminated by the two-character sequence of a carriage return followed by a line feed (CRLF). Both ends must convert between this format and whatever the local host uses as a line delimiter.

An octet transfer treats the data as 8-bit bytes with no interpretation.

Each data packet contains a block number that is later used in an acknowledgement packet. When reading a file the client sends an RRQ specifying the filename and mode as one command.

If the client can read the file, the server responds with a data packet with a block number of one. The client sends an ACK of block number one. The server responds with the next data packet with a block number of two. The client sends an ACK of block number two. This continues until the file is transferred.

In the case of a WRQ, the client specifies the filename and mode. If the client can write the file, the server responds with an ACK of block number zero. The client then sends the first 512 bytes of file with a block number of one. The server responds with an ACK of block number one.

The final TFTP message type is the error message, with an opcode of five. This is what the server responds with if an RRQ or WRQ can't be processed. Read and write errors during file transmission also cause this message to be sent, and transmission is then terminated. The error number gives a numeric error code, followed by an ASCII error message that might contain additional, operating system-specific information.

If the sender times out waiting for an ACK, it retransmits the last unacknowledged data block. If the receiver times out waiting for a data block, it retransmits the last ACK block. Both the client and server have preset values defined for the number of retransmits and the time interval between retransmits.

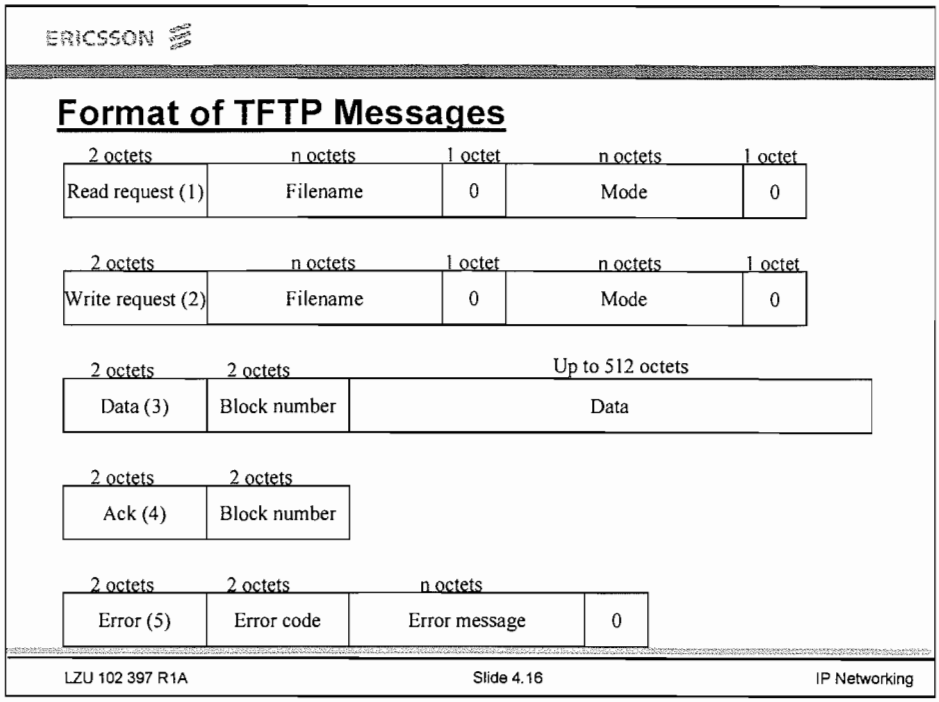


Figure 4-15.

Notes:

Simple Mail Transfer Protocol (SMTP)

The function of a mail program is to facilitate the exchange of electronic messages between users on a network. The Internet uses SMTP in order to provide a mail service.

SMTP is an applications layer program that interfaces directly with the transport layer TCP. The destination, well-known port number 25, is used in the TCP header, which causes TCP to direct incoming segments to SMTP for processing.

In this section the components involved in the SMTP process and the commands and replies used by SMTP are examined. The operation of the SMTP protocol is also illustrated with an example.

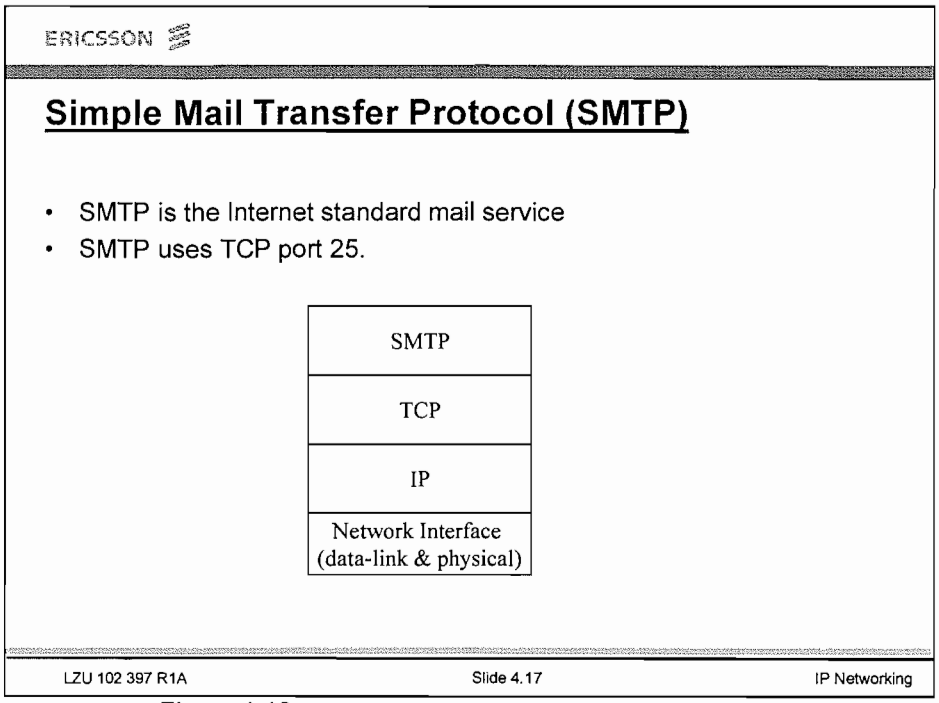


Figure 4-16.

Notes:

User Message Preparation

The message, called memo, is typically sent interactively by a user. The portion of the client that interfaces with the user is called the agent.

The agent is a user-friendly program that accepts the minimum required elements of the message in two parts, a header part and a message part.

The header part contains answers to prompts for required fields, 'To', 'From' and 'Date' and other optional fields such as 'Sender', 'Message-ID', 'Reply To', and CC'. The message part is simply text.

User Agent Functions

In addition to assisting the user with message input (prompts), the agent builds a formatted message (US ASCII characters only) with the information supplied by the user. The formatted message header consists of a variable number of header lines. The header is terminated by a blank line '(CRLF)(CRLF)' and followed by the message as composed by the user.

The user agent then builds a standardised list of required destinations (derived from user inputs and converted to machine-readable format) and sends both the standardised list and the formatted message, tagged together, to a queue for the SMTP client.

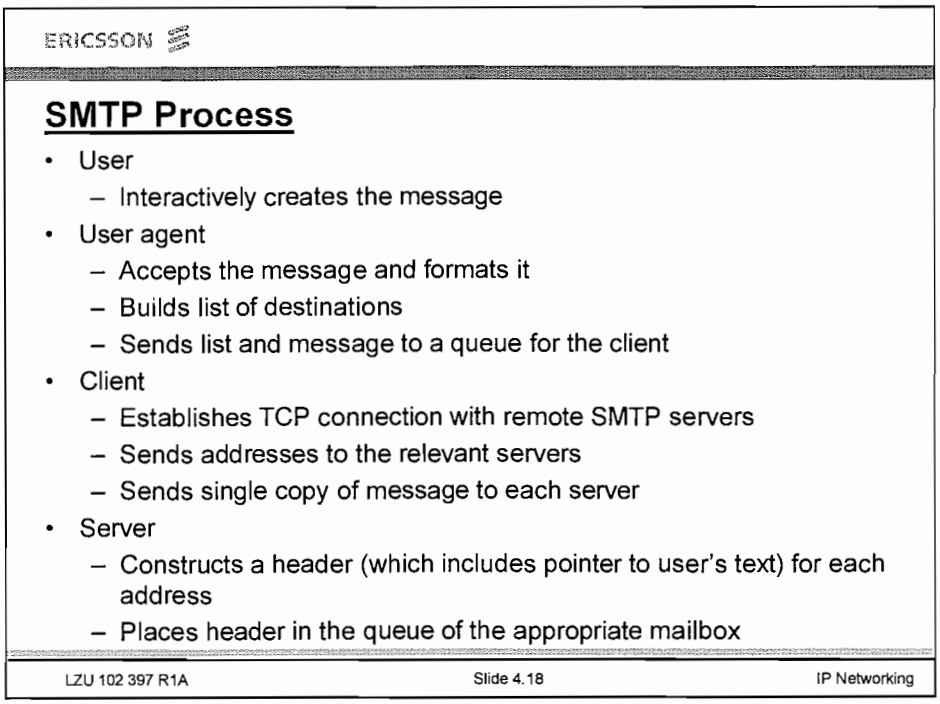


Figure 4-17.

Notes:

SMTP Client Functions

The SMTP client reads the next message from the queue of user agent inputs. If an address in the standardised list is for this SMTP system, it is handed off directly to the SMTP server. Otherwise a TCP connection is established with each remote SMTP server machine indicated by the addresses in the standardised list.

The SMTP client sends each address in the standardised list, one at a time, to the SMTP server associated with the address. The SMTP server sends an OK reply for each address received and the SMTP client marks the address in the standardised list as sent.

When all addresses in the standardised list are sent, the SMTP sends a single copy of the message to each SMTP server with a TCP connection. The SMTP server sends a positive acknowledgement for the message, and the SMTP client is no longer responsible for the message.

SMTP Server Functions

The SMTP server constructs a header for each address received and places it in the queue of the appropriate mailbox or an output queue of another SMTP server if the message is being forwarded. The header contains a pointer to the user's text and typically five lines of header constructed by the SMTP server.

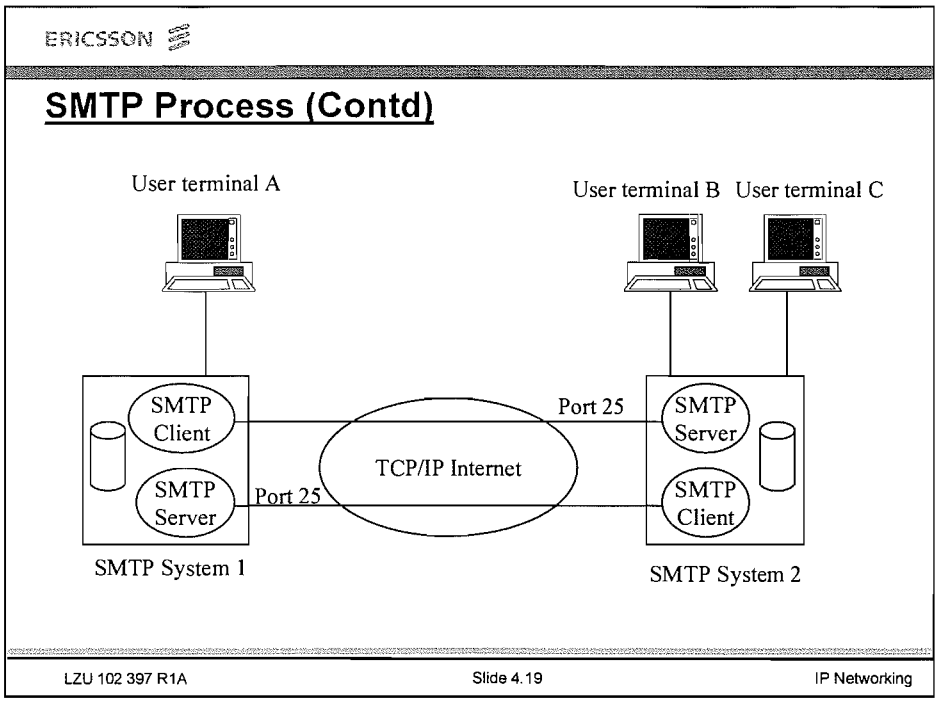


Figure 4-18.

Notes:

SMTP Commands and Replies

An SMTP command (sent by SMTP client) and reply (sent by SMTP server) consist of a string of seven-bit ASCII characters. When the transport service provides an 8-bit byte (octet) transmission channel, each 7-bit character transmitted is right justified in an octet with the high order bit cleared to zero.

The commands and replies are not case sensitive, except for the mailbox user name, forward-path names, and reverse-path names.

All commands are four characters in length and are terminated with an end-of-line (EOL), or by a space (SP) character if an argument is present. EOL is composed of two ASCII characters, carriage return (CR) and line feed (LF) and is typically illustrated as (CRLF). Most keyboards create the CRLF from the single 'ENTER' keystroke.

When a return path or forward path is part of an argument it is encoded in angle brackets (<path>), which are transmitted as part of the command. For clarity any other string or domain name is illustrated as {string} or {name} and the '{'and '}' characters are not transmitted as part of the command or reply.

The sequence of the main SMTP commands is illustrated in the diagram. The following section describes these commands.

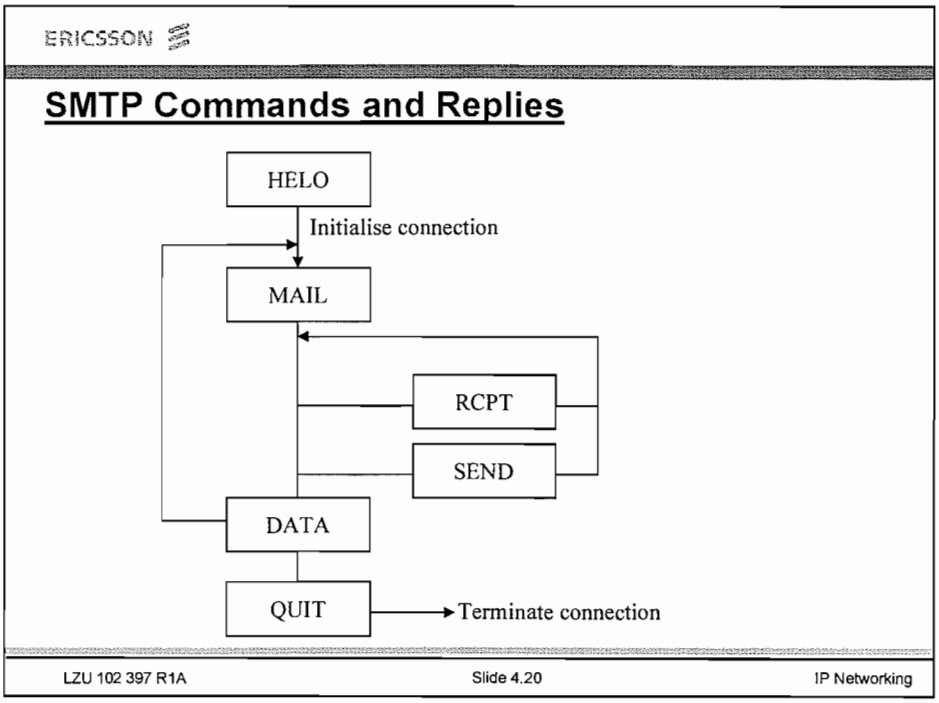


Figure 4-19.

Notes:

Hello Command (HELO)

Once the server has sent the welcoming message and the client has received it, the client normally sends the EHLO command to the server, indicating the client's identity. In addition to opening the session, use of EHLO indicates that the client is able to process service extensions and requests that the server provides a list of the extensions it supports.

Older SMTP systems which are unable to support service extensions and contemporary clients which do not require service extensions in the mail session being initiated, MAY use HELO instead of EHLO. The argument field contains the host name of the SMTP client.

The command is of the form:

'HELO (SP){domain host name}(CRLF)'.

Mail Command (MAIL)

The MAIL command is the first command in the process after connection establishment and the argument identifies the sender.

The command is of the form:

'MAIL (SP) FROM:<reverse path>(CRLF)'

The <reverse path> may be the originator only or a list of relay names with the first name being the most recent.

Recipient Command (RCPT)

The recipient command is used to identify an individual recipient of the mail. The argument consists of a destination mailbox address and is optionally preceded by a list of forward paths. The RCPT command is repeated for each intended recipient.

The RCPT command is of the form:

'RCPT (SP) TO:<forward path>(CRLF)'.

For the command to be successful, the message must be placed in queue for the destination mailbox.

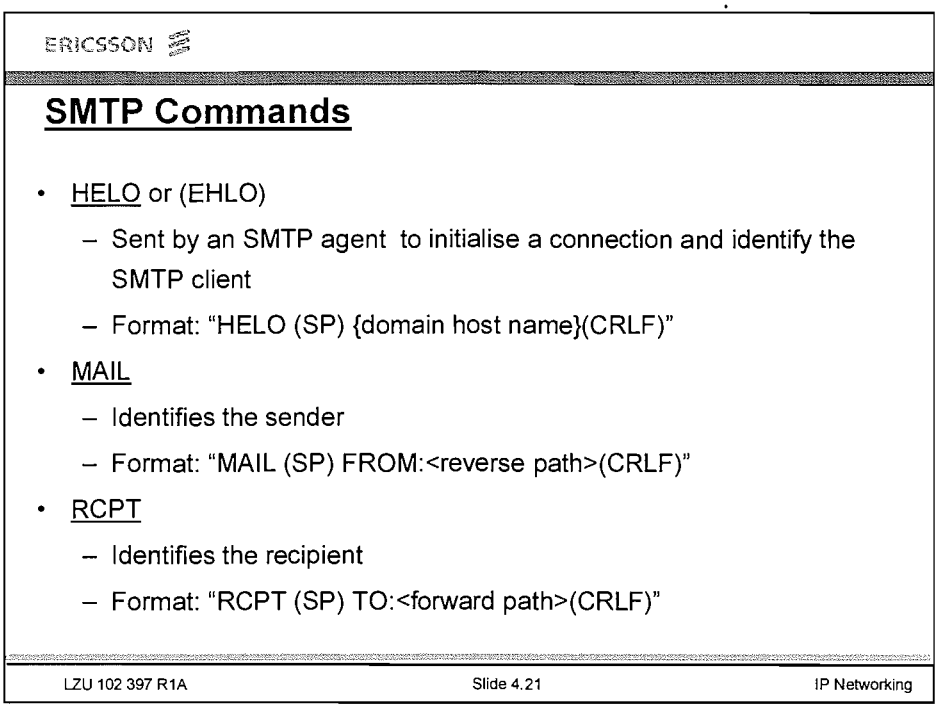


Figure 4-20.

Notes:

Data Command (DATA)

The DATA command informs the SMTP server that the phase of sending forward paths is complete and that the next communication is data.

The DATA command is of the form: 'DATA(CRLF)'.

The normal reply from the SMTP server is a 354 code with the text 'Start mail input; end with <CRLF>.<CRLF>'.

Upon receipt of the 354 code reply, the SMTP client sends the entire message (associated with the forward paths supplied in previous commands) as a string of octets. Each octet contains a seven-bit ASCII character with the parity bit set to zero.

The mail data is terminated with (CRLF).(CRLF). Upon termination of the data command, the SMTP server processes the message by placing a tag (containing a pointer to the message) in appropriate mailbox(es) and forward queue(s) if required. The SMTP server then sends an OK reply (code 250) to the SMTP client.

Send Command (SEND)

The SEND command is used to identify an individual terminal to receive the mail. The command is the same as the RCPT, except that the message is being sent to a terminal instead of a mailbox. The RCPT command is of the form:

'SEND (SP) TO:<reverse path>(CRLF)'.

The command is successful if the message is delivered to the specific terminal.

Quit Command (QUIT)

The QUIT command advises the SMTP server that the SMTP client is finished and that it should close the channel. The QUIT command is of the form: 'QUIT(CRLF)'.

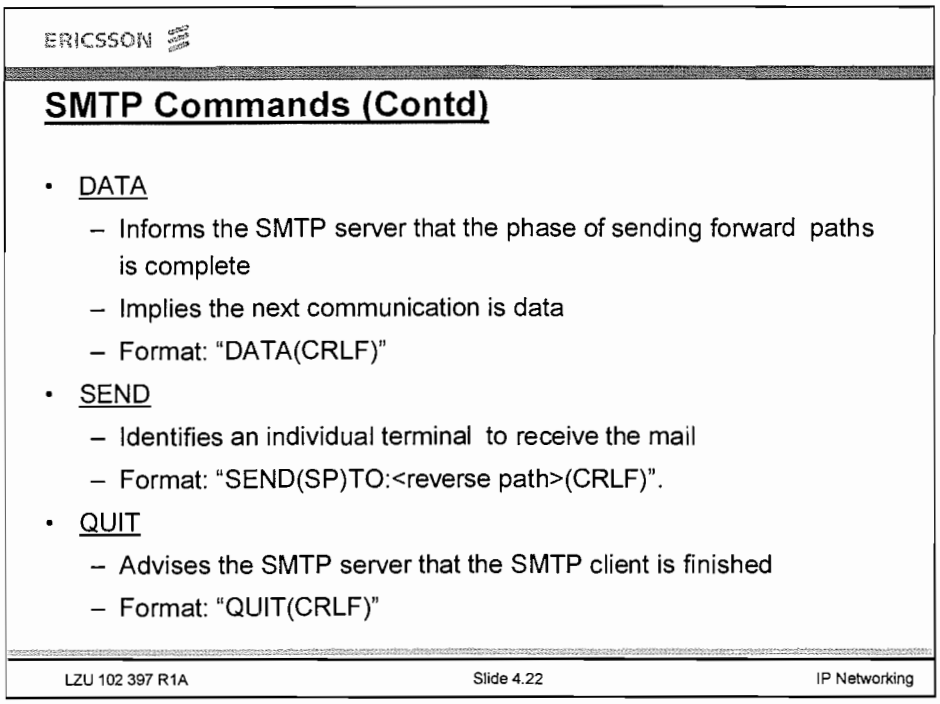


Figure 4-21.

Notes:

Reply Codes

Replies are always sent by the SMTP server in response to an SMTP client command. The reply consists of a three-digit code, a space code (SP), and a textual description of the code. The diagram identifies all SMTP server reply codes with a brief description of each. The reply is of the form: 'nnn(SP){message text}'.

There are three classes of reply codes: standard, fail and error.

- A standard code indicates everything is okay. For example, the domain address the client supplied was located - code 250.
- An error code indicates an error, but the session may continue. For example, the domain address the client supplied was not located - code 500.
- A fail code indicates the inability to continue. For example, requested action aborted due to insufficient storage - code 452.

ERICSSON

Reply Codes

Reply Code	Meaning
211	System status
214	Human information about how to use SMTP
220	<domain> service ready
221	<domain> service closing channel
250	Requested mail action okay, completed
251	User not local, forwarded to forward path
354	Start mail input, end with <CRLF>.<CRLF>
421	<domain> Service not available
450	Requested action aborted; mailbox unavailable
451	Requested action aborted; error in processing
452	Requested action aborted; insufficient storage

Reply Code	Meaning
500	Syntax error, command unrecognised
501	Syntax error, in parameters or arguments
502	Command not implemented
503	Bad sequence number
504	Command parameter not implemented
550	Requested action not taken; mailbox unavailable
551	Requested action not taken; error in processing
552	User not local; please try <forward path>
553	Action not taken; mailbox name not allowed
554	Transaction failed

LZU 102 397 R1A Slide 4.23 IP Networking

Figure 4-22.

Notes:

SMTP Protocol Example

The diagram contains an example of the SMTP protocol. The first column is a reference number.

1. The SMTP server listens on port number 25 and when the TCP connection with the SMTP client is made, it sends a greeting message that indicates readiness and identifies itself with the official name of the service host. The number 220 in the greeting message is the SMTP code for 'the system is ready'. After this initial poll, the SMTP client is master of the synchronous protocol.
2. The SMTP client sends an hello (HELO) command with its domain host name for validation. It is analogous to a person saying 'Hello. My name is Jack. Will you accept my mail?'
3. The SMTP server sends a code 250 reply (OK) with its domain name. Alternatively, it could be any one of six error codes.
4. The SMTP client sends a MAIL FROM command, which tells the SMTP server to initialise for new mail by resetting the state tables. The command also gives the SMTP server the reverse path (Smith@Test.sys) that can be used for addressing error messages resulting from this mail command.
5. The SMTP server sends an OK reply (code 250).
6. The SMTP client sends a RCPT TO command, which sets up the server with one recipient, which is 'Jones', at mail station 'Mfg.tst'.
7. The SMTP server sends an OK reply (code 250).

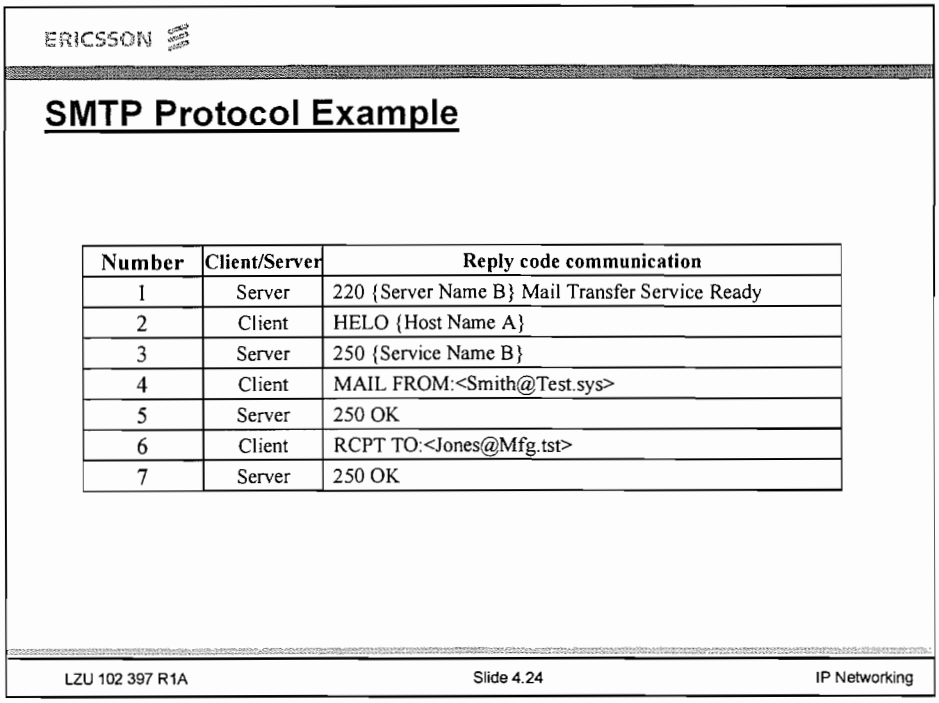


Figure 4-23.

Notes:

SMTP Protocol Example (Cont)

8. The SMTP client sends another RCPT TO command, except with a different <forward path>. This destination name is invalid.
9. The SMTP server sends a reply code 550, which means 'No such user here'.
10. The SMTP client sends another RCPT TO command with the forward path of <Bob@Engr.dev>.
11. The SMTP server accepts the forward path <Bob@Engr.dev> and returns an OK reply (code 250).
12. The SMTP client has completed the set-up of all forward paths associated with this message and sends a DATA command, which alerts the SMTP server of data to come.
13. The SMTP server sends the reply, 'Start mail input; end with '<CRLF>.<CRLF>'. The go-ahead reply code is 354.
14. The SMTP client sends the entire memo (string of ASCII characters supplied by the user).
15. The SMTP client sends the data terminate command, '<CRLF>.<CRLF>'.
16. The SMTP server replies with an OK (code 250).
17. The SMTP client sends the QUIT command to terminate the SMTP server connection.
18. The SMTP server places a tag in each mailbox of Smith, Jones and Bob. The server sends a reply 221 that contains its service name and the message 'Service closing channel'.

ERICSSON

SMTP Protocol Example (Contd)

Number	Client/Server	Reply code communication
8	Client	RCPT TO:<Williams@Mfg.tst>
9	Server	550 No such user here
10	Client	RCPT TO:<Bob@Engr.dev>
11	Server	250 OK
12	Client	DATA
13	Server	354 Start mail input; end with <CRLF>.<CRLF>
14	Client	{ASCII character text}
15	Client	<CRLF>.<CRLF>
16	Server	250 OK
17	Client	QUIT
18	Server	221 {Host Name B} Service closing channel

LZU 102 397 R1A Slide 4.25 IP Networking

Figure 4-24.

Notes:

POST OFFICE PROTOCOL VERSION 3 (POP3)

Typically user terminals, for example PCs, do not have sufficient resources (cycles, disk space) in order to permit a SMTP server and associated local mail delivery system to be kept resident and continuously running.

User terminals support a user agent (UA) to aid the tasks of mail handling. Servers that support SMTP offer a mail-drop service to the user terminals.

The Post Office Protocol, Version 3 (POP3), is intended to permit a client to dynamically access a mail-drop on a server in a useful fashion. Usually, this means that the POP3 protocol is used to allow a client to retrieve mail that the server is holding for it.

POP3 is not intended to provide extensive manipulation operations of mail on the server. Normally, mail is downloaded and then deleted.

A more advanced (and complex) protocol, IMAP4, is discussed later in the text.

When the user agent on a client wishes to enter a message into the transport system, it establishes an SMTP connection to its relay host and sends all mail to it. This relay host could be, but need not be, the POP3 server host for the client host.

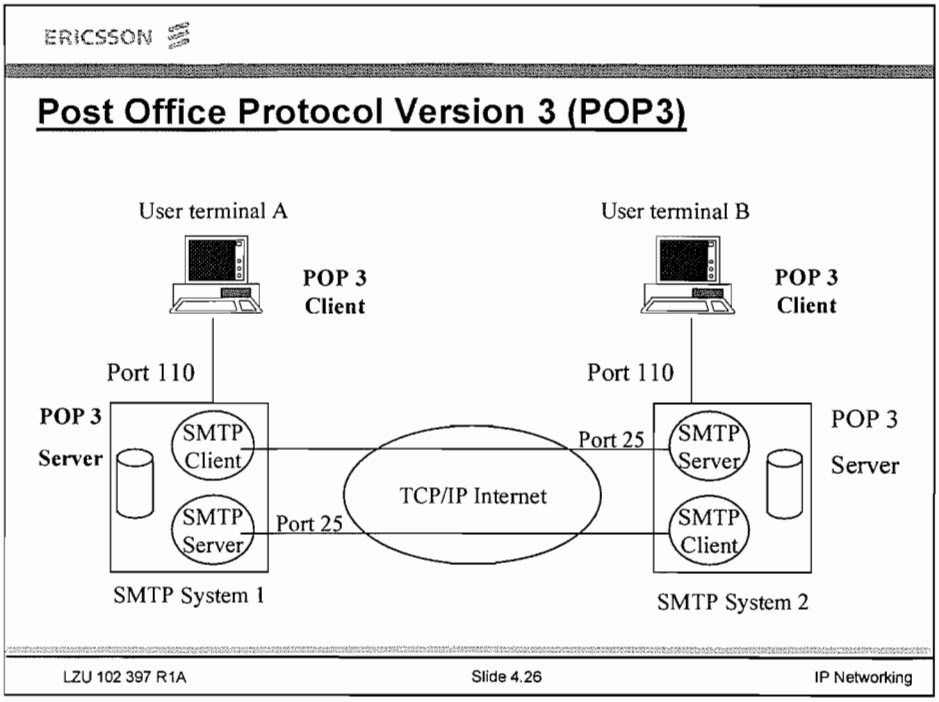


Figure 4-25.

Notes:

POP3 Basic Operation

Initially, the server starts the POP3 service by listening on TCP port 110. When a client wishes to make use of the service, it establishes a TCP connection with the server. When the connection is established, the POP3 server sends a greeting. The client and POP3 server then exchange commands and responses (respectively) until the connection is closed or aborted.

Commands in the POP3 consist of a case-insensitive keyword, possibly followed by one or more arguments. All commands are terminated by a CRLF.

Keywords and arguments consist of printable ASCII characters. Keywords and arguments are each separated by a single space character. Keywords are three or four characters long. Each argument may be up to 40 characters long.

Responses in the POP3 consist of a status indicator and a keyword possibly followed by additional information. All responses are terminated by a CRLF. Responses may be up to 512 characters long, including the terminating CRLF.

There are currently two status indicators: positive ("+OK") and negative ("-ERR"). Servers must send the "+OK" and "-ERR" in upper case. Responses to certain commands are multi-line. In these cases, after sending the first line of the response and a CRLF, any additional lines are sent, each terminated by a CRLF.

When all lines of the response have been sent, a final line is sent, consisting of a termination octet (period or .) and a CRLF. If any line of the multi-line response begins with the termination octet, the line is "byte-stuffed" by pre-pending the termination octet to that line of the response. Hence a multi-line response is terminated with the five octets "(CRLF).(CRLF)".

When examining a multi-line response, the client checks to see if a line begins with the termination octet. If so and if octets other than CRLF follow, the first octet of the line (the termination octet) is stripped away. If CRLF immediately follows the termination octet, then the response from the POP server is ended and the line containing ".CRLF" is not considered part of the multi-line response.

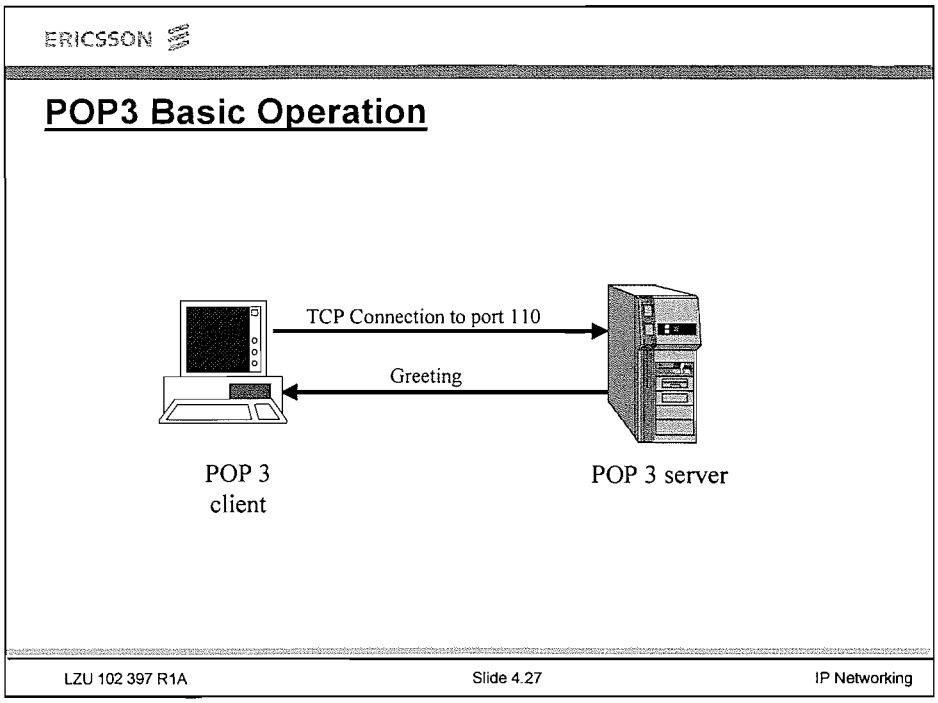


Figure 4-26.

Notes:

POP3 Basic Operation (Contd)

A POP3 session progresses through a number of states during its lifetime. Once the TCP connection has been opened and the POP3 server has sent the greeting, the session enters the Authorisation State. In this state, the client must identify itself to the POP3 server.

Once the client has successfully done this, the server acquires resources associated with the client's mail-drop, and the session enters the Transaction state. In this state, the client requests actions on the part of the POP3 server. When the client has issued the Quit command, the session enters the Update State. In this state, the POP3 server releases any resources acquired during the Transaction State and says goodbye. The TCP connection is then closed.

A server must respond to an unrecognised, unimplemented, or syntactically invalid command with a negative status indicator. A server must respond to a command issued when the session is in an incorrect state with a negative status indicator.

There is no general method for a client to distinguish between a server that does not implement an optional command and a server that is unwilling or unable to process the command.

A POP3 server may have an inactivity auto-logout timer. Such a timer must be of at least 10 minutes' duration. The receipt of any command from the client during that interval should suffice to reset the auto-logout timer. When the timer expires, the session does not enter the Update State; the server should close the TCP connection without removing any messages or sending any response to the client.

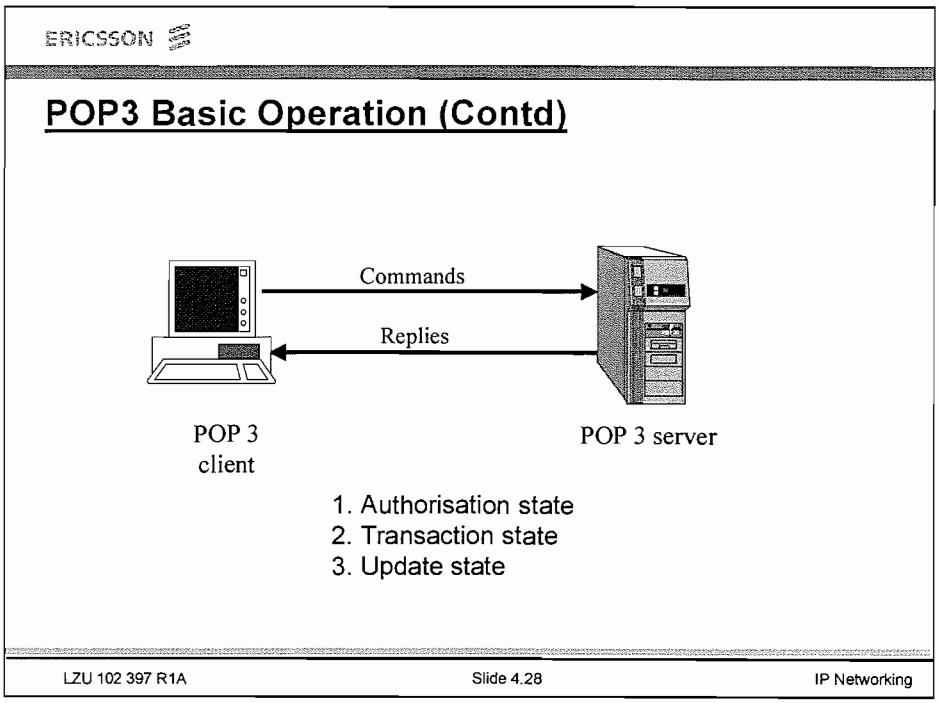


Figure 4-27.

Notes:

POP3 Commands

USER name (name of mail box)

To authenticate using the USER and PASS command combination, the client must first issue the USER command. If the POP3 server responds with a positive status indicator ("+OK"), then the client may issue either the PASS command to complete the authentication, or the QUIT command to terminate the POP3 session. If the POP3 server responds with a negative status indicator ("-ERR") to the USER command, then the client may either issue a new authentication command or may issue the QUIT command.

PASS name (password of mailbox)

When the client issues the PASS command, the POP3 server uses the argument pair from the USER and PASS commands to determine if the client should be given access to the appropriate mail-drop. Since the PASS command has exactly one argument, a POP3 server may treat spaces in the argument as part of the password, instead of as argument separators.

QUIT

When the client issues the QUIT command, during the Authorisation state, the session enters the Update state.

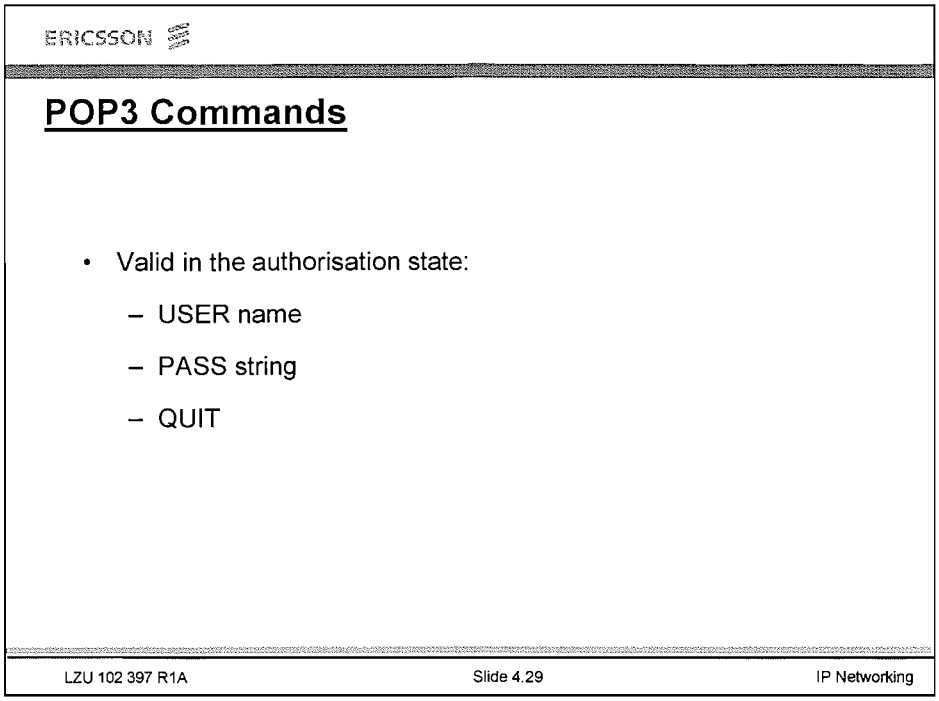


Figure 4-28.

Notes:

POP3 Commands (Contd)

STAT

The POP3 server issues a positive response with a line containing information for the mail drop. This line is called a "drop listing" for that mail drop. In order to simplify parsing, all POP3 servers are required to use a certain format for drop listings. The positive response consists of "+OK" followed by a single space, the number of messages in the mail drop, a single space, and the size of the mail drop in octets.

LIST (message number)

If an argument was given and the POP3 server issues a positive response with a line containing information for that message. This line is called a "scan listing" for that message. If no argument was given and the POP3 server issues a positive response, then the response given is multi-line. After the initial +OK, for each message in the mail drop, the POP3 server responds with a line containing information for that message. This line is also called a "scan listing" for that message. If there are no messages in the mail drop, then the POP3 server responds with no scan listings. Instead, it issues a positive response followed by a line containing a termination octet and a CRLF pair.

RETR message number

If the POP3 server issues a positive response, then the response given is multi-line. After the initial +OK, the POP3 server sends the message corresponding to the given message-number, being careful to byte-stuff the termination character (as with all multi-line responses).

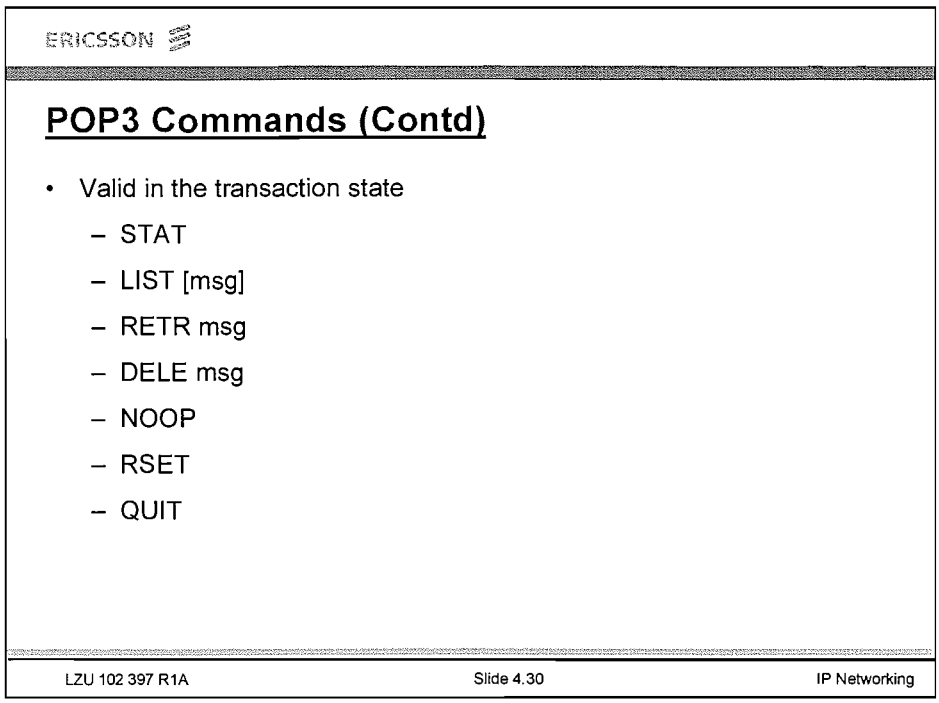


Figure 4-29.

Notes:

POP3 Commands (Contd)

DELETE message number

The POP3 server marks the message as deleted. Any future reference to the message-number associated with the message in a POP3 command generates an error. The POP3 server does not actually delete the message until the POP3 session enters the UPDATE state.

NOOP

The POP3 server does nothing. It merely replies with a positive response.

RSET

If any messages have been marked as deleted by the POP3 server, they are unmarked. The POP3 server then replies with a positive response.

QUIT

The POP3 server removes all messages marked as deleted from the mail drop and replies as to the status of this operation. If there is an error, such as a resource shortage, encountered while removing messages, the mail drop may cause some or none of the messages marked as deleted to be removed. In no case may the server remove any messages not marked as deleted. Whether the removal was successful or not, the server then releases any exclusive-access lock on the mail-drop and closes the TCP connection.

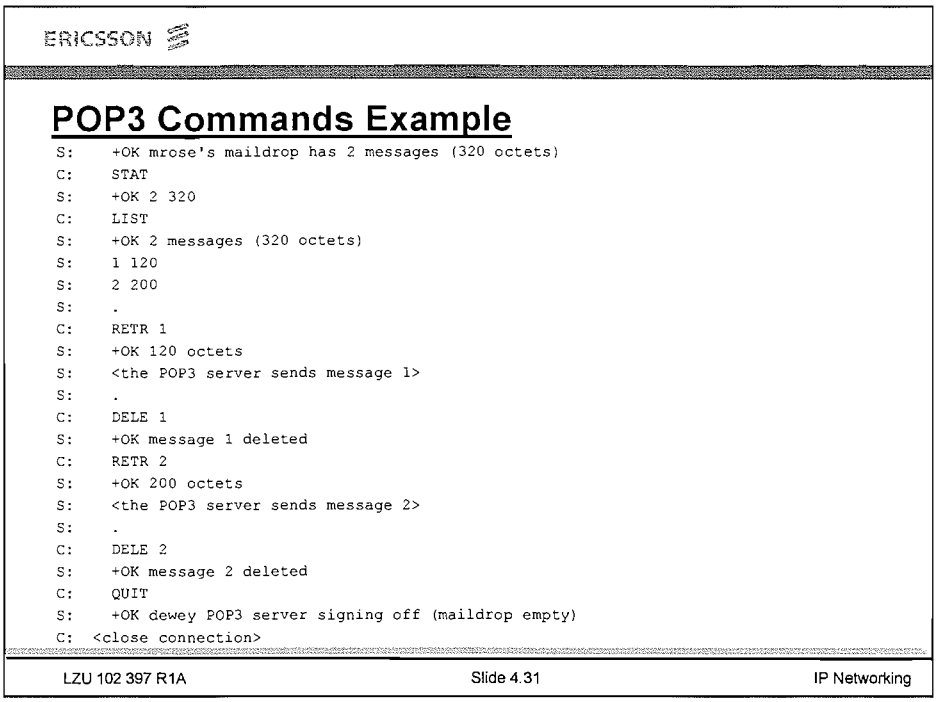


Figure 4-30.

Notes:

INTERNET MESSAGE ACCESS PROTOCOL, VERSION 4 (IMAP4)

The Internet Message Access Protocol, Version 4 (IMAP4) allows a client to access and manipulate electronic mail messages on a server. IMAP4 permits manipulation of remote message folders, called "mailboxes", in a way that is functionally equivalent to local mailboxes. IMAP4 includes operations for creating, deleting, and renaming mailboxes; checking for new messages; permanently removing messages; setting and clearing flags; searching; and selective fetching of message attributes, texts, and portions thereof. Messages in IMAP4 are accessed by the use of numbers. These numbers are either message sequence numbers (relative position from 1 to the number of messages in the mailbox) or unique identifiers (immutable, strictly ascending values assigned to each message, but which are not necessarily contiguous).

IMAP4 does not specify a means of posting mail; this function is handled by a mail transfer protocol such as SMTP.

The IMAP4 protocol assumes a reliable data stream such as that provided by TCP. When TCP is used, an IMAP4 server listens on port 143.

Commands and Responses

- An IMAP4 session consists of the establishment of a client-server connection, an initial greeting from the server, and client-server interactions. These client-server interactions consist of a client command, server data, and a server completion result response.

- All interactions transmitted by client and server are in the form of lines, that is, strings that end with a CRLF. The protocol receiver of an IMAP4 client or server is either reading a line, or is reading a sequence of octets with a known count followed by a line.

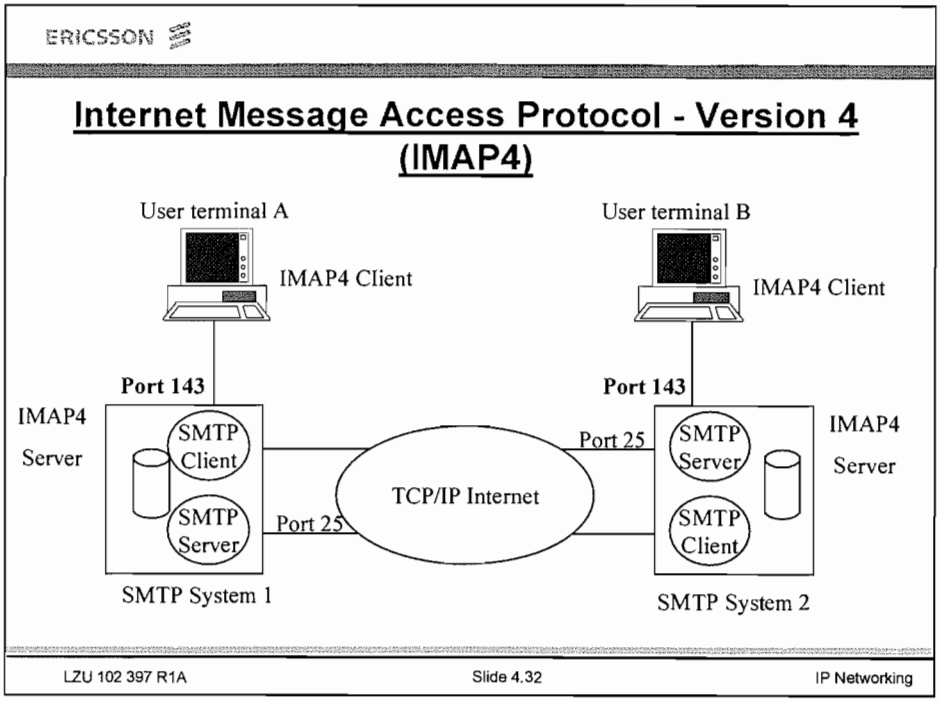


Figure 4-31.

Notes:

HYPERTEXT TRANSFER PROTOCOL (HTTP)

The Internet standard communication protocol between Web servers and clients is the Hypertext Transfer Protocol (HTTP). HTTP uses the well-known TCP port 80.

The Internet standard language for writing Web documents is HTML (Hypertext Markup Language).

To create a Web document you must use the HTML tags to build the logical structure of the document, for example headers, lists and paragraphs.

All documents, images, audio or video clips on the Web are called resources. To address and identify the access method for these resources the Web uses URLs (Uniform Resource Locators).

URL is an Internet standard tracking protocol and can be found under RFC 1738. Every Web page is assigned a unique URL (Uniform Resource Locator) that effectively serves as the page's world-wide name. URLs have three parts: the protocol, the DNS name of the host on which the page is located, and a local name uniquely indicating the specific page (usually a file name and path). For example, http://www.ericsson.com/datacom/solutions.

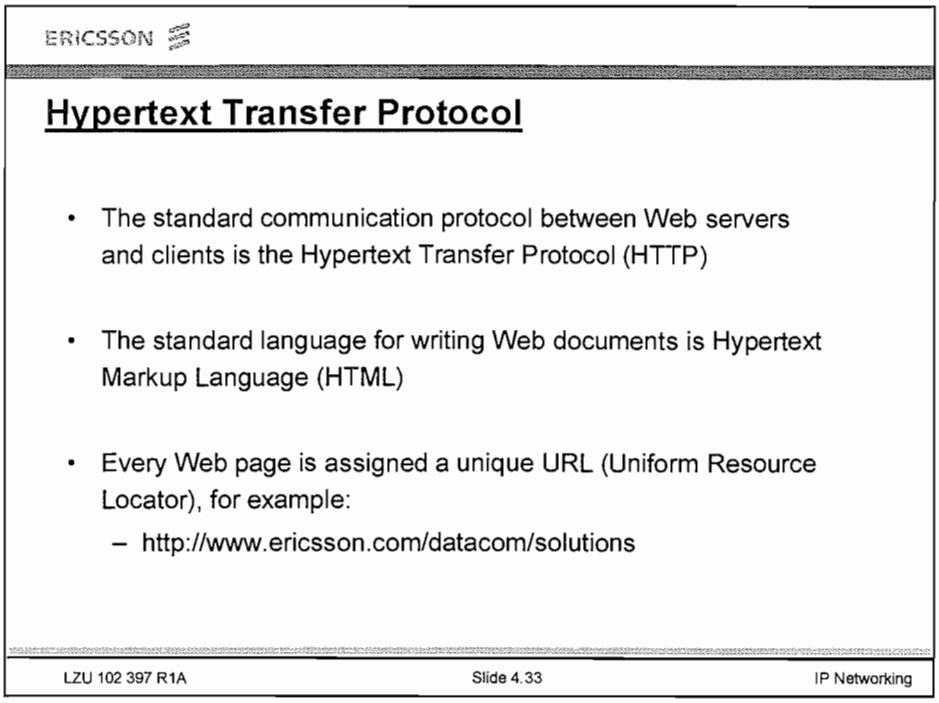


Figure 4-32.

Notes:

HTTP Message Formats

The request message is an ASCII character string containing a verb (command describing what to do) and an address (where to do it at). The address format is defined by a uniform resource location (URL), which also includes the protocol to be used. The ASCII string of a general HTTP command from the client to the server appears as:

GET http://server.name/path/file.type

The verb "GET" is a solicitation for service from the client to the server. The remainder constitutes the URL.

"HTTP://" defines the protocol to be used (it could also be FTP, SMTP, NNTP and so on).

The next field is the name of the HTTP server containing the requested data, which is followed by the path name and file name separated by a forward slash (/). When a forward slash follows the server name without a path and file name, the default file is requested. This is typically the home page of the HTTP server.

The response message contains the requested data and then the server closes the connection. Although new fields have been added to revised versions of HTTP, this basic format is still accepted for backward compatibility.

HTTP revision 1 (HTTP/1.0) began the standardisation of HTTP by defining header parts and body parts similar to SMTP mail. The main changes were to add two new verbs (HEAD and POST) called methods, and to formally identify the version of HTTP with a major and minor number.

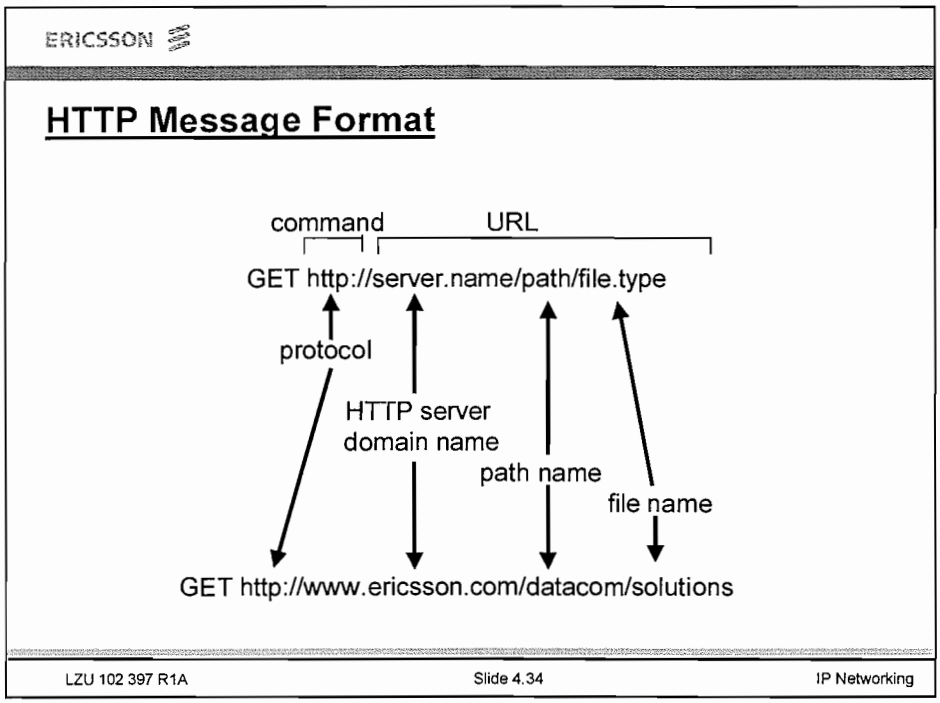


Figure 4-33.

Notes:

HTTP/1.1 Full Request

To differentiate between HTTP/1.1 and earlier versions, the original request is called a simple request and the HTTP/1.1 version is called a full request.

The full request may have three different header parts and a body part separated by a blank line. That is, the last header part is ended with a double (CR)(LF) sequence, which is the same as SMTP.

The full request format is:

Method(SP)http://server.name/path/file.type(SP)HTTP/1.0(CR)(LF)
General-header (CR)(LF)

Request-header (CR)(LF)

Entity-header (CR)(LF) (CR)(LF)

Body

The methods are described as follows

GET: The GET method retrieves the data identified by the URL address. If the content of the URL address is a process, the data from that process is returned.

HEAD: The HEAD method is identical to the GET method except the response contains only the header (general, response and entity). This provides information about the source of information and is used for testing the validity of hypertext links. From the response message, a user may determine information about the web page, such as the last date modified.

POST: The POST method allows the end-user to place data on the server. This may be used in a bulletin board operation, when submitting a form for registration, or to enhance an existing database.

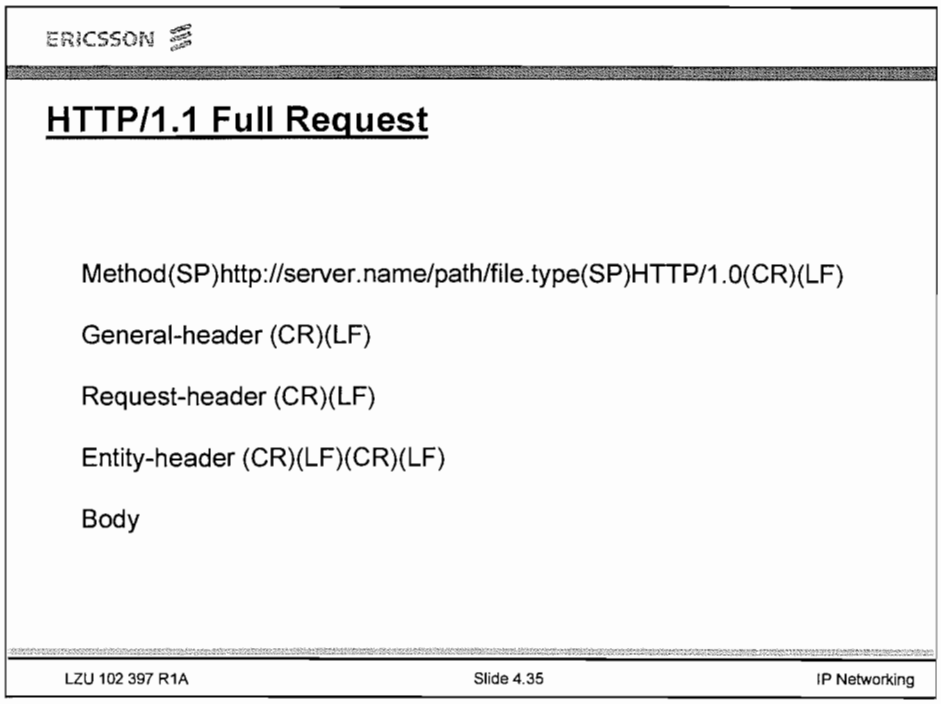


Figure 4-34.

Notes:

Web Operation Example

In the example illustrated, assume that the user has just clicked on some piece of text or perhaps an icon that points to the page whose URL is http:// www.w3.org/hypertext/WWW/TheProject.html. The three parts of the URL are: the name of the protocol (http), the name of the server where the page is located (www.w3.org), and the name of the file containing the page (hypertext/WWW/TheProject.html).

The steps that occur between the user's click and the page being displayed are as follows:

1. The browser determines the URL (by seeing what was selected).
2. The browser asks DNS for the IP address of www.w3.org.
3. DNS replies with the IP address of this server, 18.23.0.23.
4. The browser makes a TCP connection to port 80 on 18.23.0.23.
5. It then sends a GET /hypertext/WWW/TheProject.html command.
6. The www.w3.org server sends the file TheProject.html.
7. The TCP connection is released.
8. The browser displays all the text in TheProject.html.
9. The browser fetches and displays all images in TheProject.html.

For each in-line image (icon, drawing, photo and so on) on a page, the browser establishes a new TCP connection to the relevant server to fetch the image. This is not efficient, but it keeps the implementation simple.

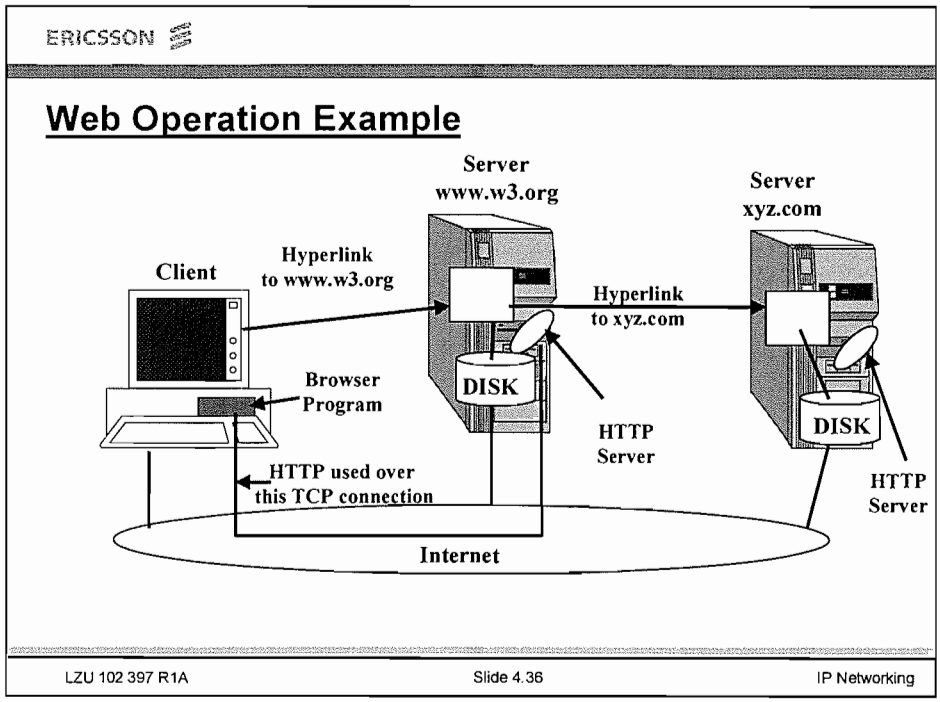


Figure 4-35.

Notes:

Intentionally Blank

5 Bridging & Switching

After completing this chapter you will be able to:

- Describe how Transparent Bridges operate
- Outline the advantages and disadvantages of Bridging
- Describe Spanning Tree Protocol
- Describe the fundamentals of LAN switches

Intentionally Blank

REPEATERS, BRIDGES AND SWITCHES

The figure shows the 7 layers of the OSI model. Each layer operates independently of the others using a method referred to as encapsulation. At the sending device each layer receiving data from the layer above will process the data, add its own protocol header and transfer the data block to the layer below. The layer below will simply treat the data as a data block, it will not try to understand its meaning. The block will be processed by the layer, which adds its own protocol header and then passes the larger data block to the layer below.

At the receiving device the reverse happens. When the data arrives, the first layer processes its peer header and then passes the data to the layer above, which carries out the same action. Ultimately, the application data originally sent by the sending device will arrive at the receiving application.

Routers operate at the network layer. They connect networks into inter-networks that are physically unified, but in which each network retains its identity as a separate network environment.

Repeaters operate at the Physical layer. They receive transmissions (bits) on a LAN segment and regenerate the bits to boost a degraded signal and extend the length of the LAN segment.

Bridges operate at the Data link layer. They connect network environments into logical and physical single inter-networks.

IEEE standard 802.3 and Ethernet

The IEEE 802.3 standard is for a CSMA/CD LAN. When a station wants to transmit, it listens to the cable. If the cable is busy, the station waits until it is idle; otherwise, it transmits immediately. If two or more stations begin to transmit simultaneously on an idle cable, they collide. All collision stations then terminate their transmission, wait a random time, and repeat the whole process all over again. The 802.3 standard specifies several physical media, such as coaxial cable for 10Base5 (thick Ethernet), 10Base2 (thin coaxial cable), 10Base-T (twisted pair) and 10Base-F (fiber). The 803.2 standard also specifies the 802.3 MAC sublayer protocol and the standards for a switched 802.3 LAN.

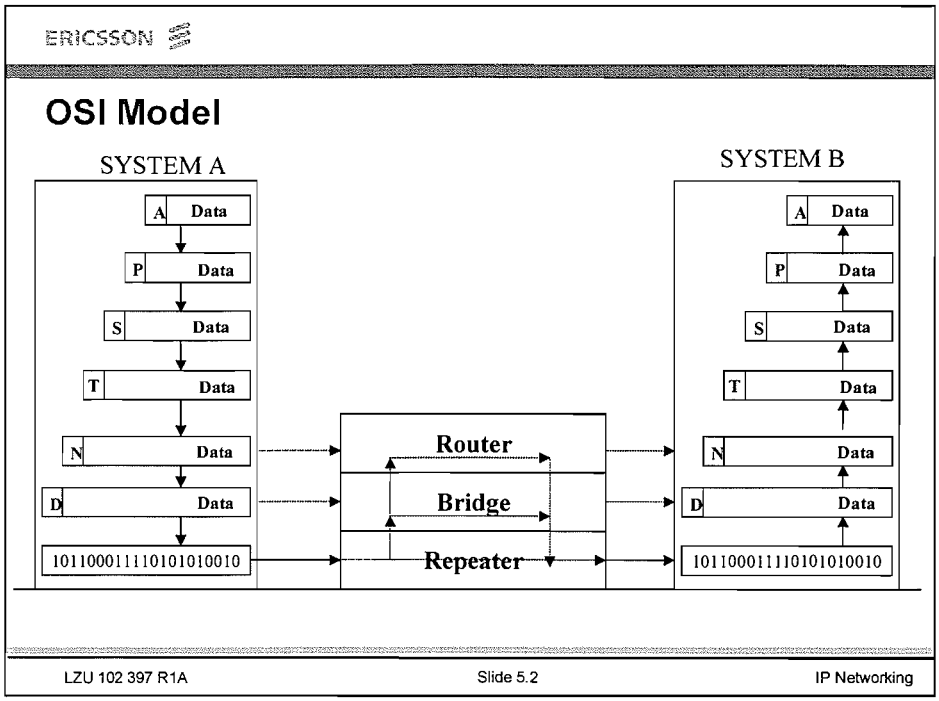


Figure 5-1.

Notes:

COLLISION DOMAIN & BROADCAST DOMAIN

To understand one of the key differences between internetworking products it is essential to appreciate what a collision domain and a broadcast domain is and the effect that each of the products has on these domains.

Collision Domain

If two devices within the domain attempt to transmit simultaneously the packets will collide and retransmission will occur.

Broadcast Domain

If a device sends out a Network layer broadcast, for example, ARP request, all devices within the same broadcast domain will receive it.

Repeaters only regenerate the signal. They do not in anyway reduce network collisions or broadcast traffic.

Bridges and Switches reduce the number of collision on the network by breaking the network into smaller segments. Two devices on either side of a bridge can put traffic on the LAN simultaneously and they will never collide with each other.

Note: A LAN switch is effectively a high-speed bridge and the some details in this chapter apply to both devices. Essentially an Ethernet Bridge has at least one Ethernet port and one WAN port, whereas a Switch has Ethernet ports only. An Ethernet switch also operates faster as it typically deals with Layer 2 only.

Routers like bridges reduce the number of collisions. In addition to this, they stop network broadcast traffic, thus reducing the amount of traffic on each segment.

A bridge is a device that connects two LAN segments. A bridge forwards complete, correct frames from one segment to another. A typical bridge consists of at least two network interfaces.

Bridges are used to span longer distances in networks. For example, a corporation may need a network that allows computers in one building to communicate with computers in another. If a significant distance separates the two buildings or if the buildings are large, a single LAN will not suffice to reach both buildings.

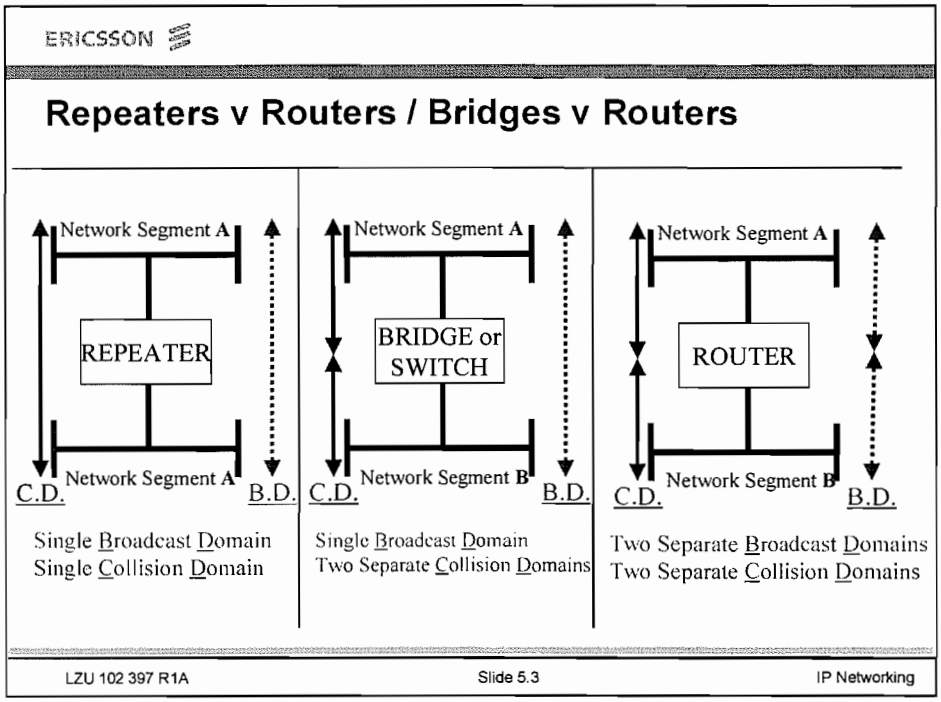


Figure 5-2.

Notes:

TRANSPARENT BRIDGING

Digital Equipment Corporation (DEC) developed Transparent Bridging. It is most often found in Ethernet networks, in which bridges pass frames along one hop at a time, based on tables associating end nodes with bridge interfaces. Transparent bridges are designed to enable frames to move back and forth between network segments running the same MAC layer protocols. It is referred to as transparent bridging because the presence of the bridges is transparent to other network devices. The bridges do not alter the data frame and the address of the bridge is never the source or destination of a frame.

Transparent Bridging Operation

There are three processes involved in transparent bridging operation. These are:

Learning

When a transparent bridge is first turned on, it knows nothing about the network topology. It learns which devices can be reached on each of its interfaces by monitoring the source MAC address of all incoming frames. It maintains a database of these learned Media Access Control (MAC) addresses and their associated interfaces in a table. The bridge updates this table every time a device sends a frame, and deletes entries of devices not heard from within a specified time period. This learning capability allows new devices to be added to the network without reconfiguring the bridge.

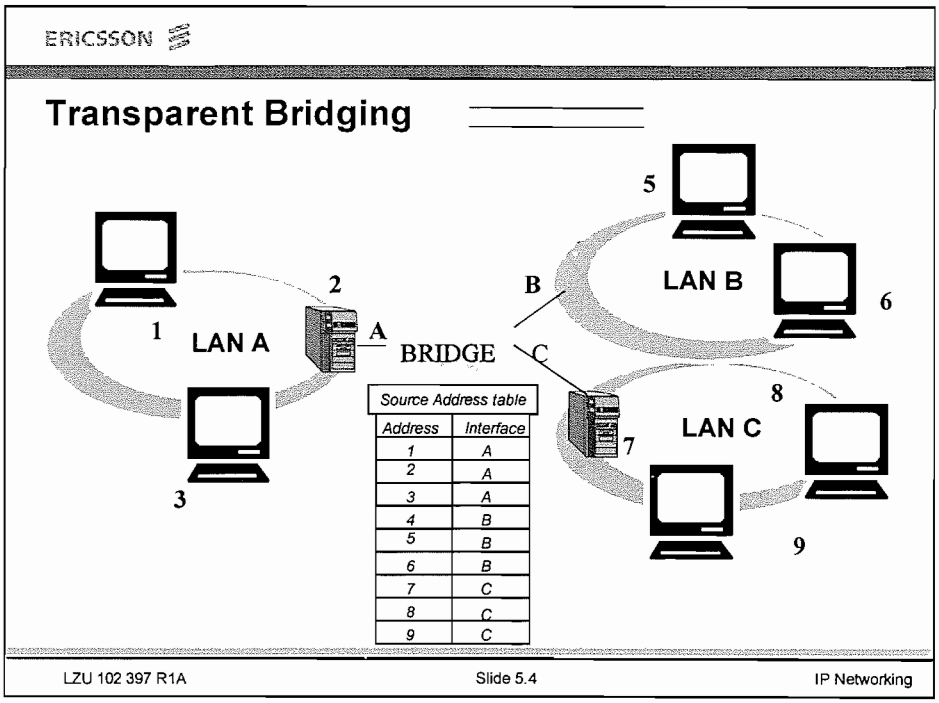


Figure 5-3.

Notes:

Forwarding

If a bridge knows where a destination address is, it forwards frames out the associated interface. If the bridge does not know where the destination address is, it forwards the frame out every interface. This is called flooding. A bridge learns addresses and forwards traffic as follows:

Assume that the source and destination addresses are located on different bridged networks, and the bridge knows neither address. The bridge notes the source address and updates its tables. It forwards the frame out all interfaces, except the one from where it was received. If a reply comes back, the bridge examines the source address, which was the original target address, and adds the entry to its table. The bridge forwards all subsequent communication between the devices.

Filtering

Bridges make a simple 'forward' or 'don't forward' decision on each frame they receive from the LAN. If a frame's destination address is on the same LAN segment as it's originating address, it is filtered out and not forwarded across the bridge. Bridges can filter frames based on any link layer field. For example, a bridge can be configured to reject all frames from a particular network. Unnecessary broadcast and multicast frames can also be filtered in this way. Data-link information often includes a reference to an upper-layer protocol, and bridges can usually filter based on this parameter too.

Device 1 on LAN A addresses a packet to device 4 on LAN B. The bridge receives this packet on Interface A and floods it out every other interface. The bridge now knows that address 1 is out interface A. The packet is received by device 4 and it replies with a packet which has a destination 1 and source 4. The bridge receives this packet on interface B, so it now knows that address 4 is out interface B. The bridge forwards the packet out interface A only, as it already knows where device 1 is. In this way, the bridge has built up and stored two entries in its source address table.

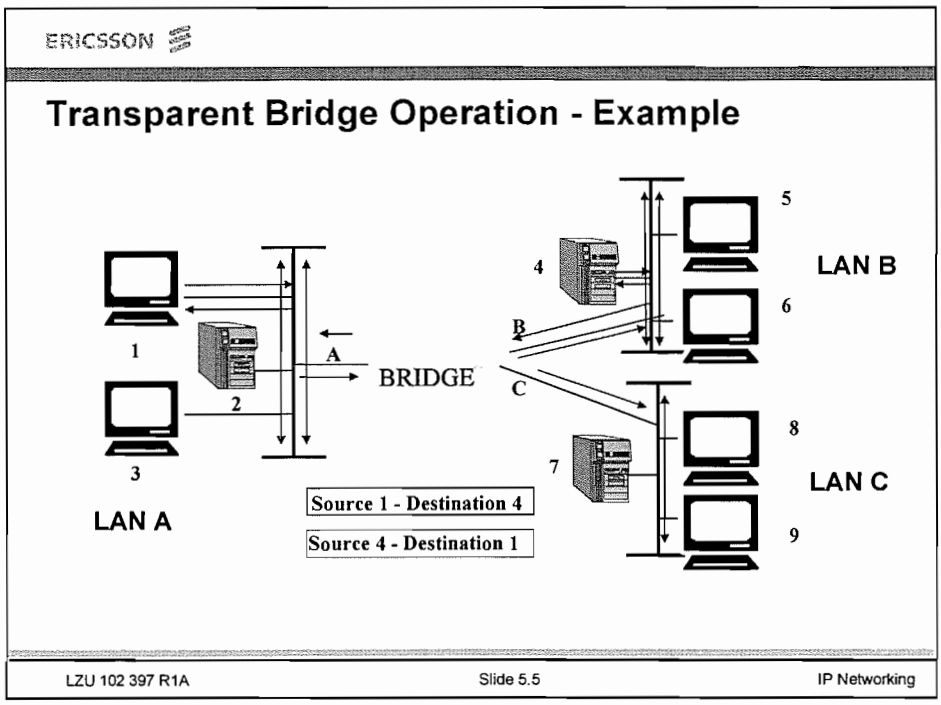


Figure 5-4.

Notes:

BRIDGING LOOPS

To increase reliability it is common practice to use two or more bridges in parallel between pairs of LANs. This arrangement, however, also introduces some additional problems because it causes loops in the topology. For example, if a packet with an unknown destination arrives at bridge 1 from LAN 1, it forwards it onto LAN 2. Bridge 2 now sees this packet on LAN 2 and, since the destination is still unknown, it forwards it onto LAN 1. Once again, bridge 1 sees the packet on LAN 1 and forwards it onto LAN 2. This cycle could go on forever, using up the bandwidth and blocking the transmission of other packets on both segments.

Preventing Loops

The Spanning Tree Protocol, sometimes referred to as the Spanning Tree Algorithm (STA), solves the problems associated with bridge loops. It allows redundant paths and ensures a loop-free topology by means of a bridge to bridge protocol. It creates this loop-free topology by blocking duplicate paths between network segments and automatically activating backup paths if a link segment or bridge fails. The STA creates a set of device-to-device paths through the network, such that there is only one active or 'primary' path between any two devices. All paths not selected by the STA are temporarily disabled. STA allows participating bridges to reactivate blocked paths if an existing primary path fails. With this feature, the STA allows networks to recover quickly and automatically if a network device, such as a bridge or a section of networking cabling fails.

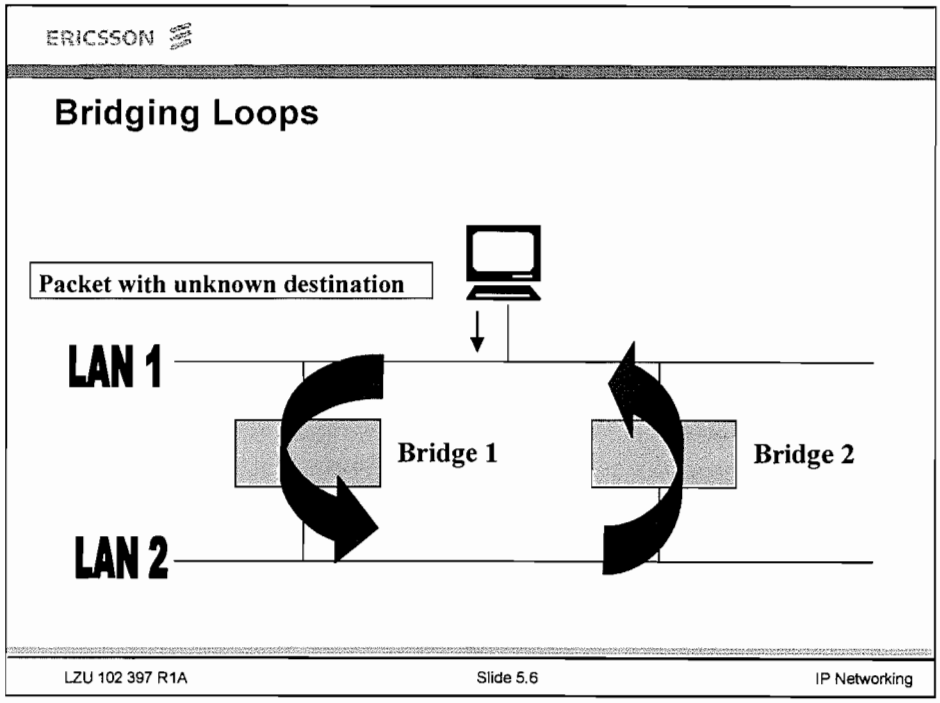


Figure 5-5.

Notes:

SPANNING TREE PROTOCOL (STP)

The Spanning Tree Protocol (STP) elects the bridge with the lowest priority to be the Root Bridge. A network administrator can configure this priority. If it is not, then the bridge with the lowest value identifier (based on the MAC address plus a priority value) becomes the root by default. Every other bridge selects the lowest-cost path to the Root Bridge. A network administrator can alter interface costs in order to select a preferred route.

All interfaces on these paths forward traffic. All interfaces not on these paths block traffic. This ensures that a unique path is established from every LAN to root. The algorithm runs continuously to detect topology changes and update the tree.

Initially, all bridges consider themselves to be the Root Bridge. Each bridge broadcasts a Bridge Protocol Data Unit (BPDU) on each of its LANs that asserts this fact. On any given LAN, only one claimant has the lowest-valued identifier and maintains this belief.

Over time, as BPDUs propagate, all bridges know the identity of the lowest-valued bridge identifier throughout the Internet. The Root Bridge regularly broadcasts the fact that it is the Root Bridge on all the LANs to which it is attached. This allows the bridges on those LANs to determine their root port and the fact that they are directly connected to the Root Bridge.

Each of these bridges in turn broadcast a BPDU on the other LANs to which it is attached (all LANs except the one on its root port), indicating that it is one hop away from the root bridge. This activity is propagated throughout the Internet. Every time a bridge receives a BPDU, it transmits BPDUs, indicating the identity of the Root Bridge and the number of hops to reach the Root Bridge.

On any LAN, the bridge claiming to be the one closest to the root becomes the designated bridge.

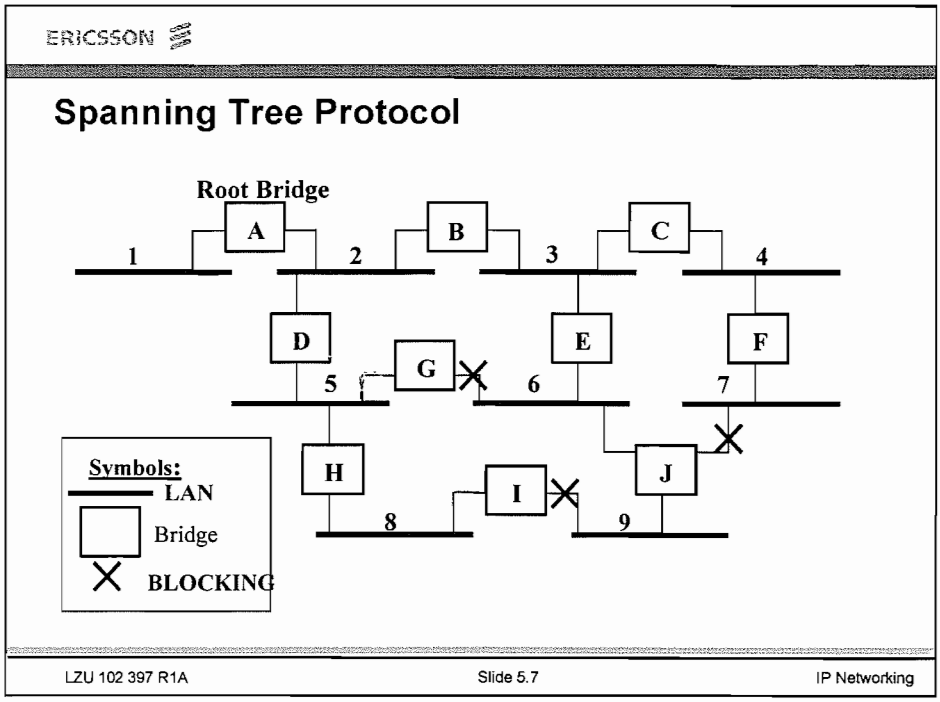


Figure 5-6.

Notes:

ADVANTAGES OF BRIDGING

Transparent bridging is connectionless and is completely invisible to the hosts and is fully compatible with all existing 802 products.

The presence of a bridge is transparent to users from the instant it is first installed, and bridges adapt automatically to network changes. Bridge based internetworks can be modified and reconfigured very easily.

Bridges can connect networks running different protocols without requiring additional software. Because bridges operate below the network layer in the OSI model, the network manager does not need to decide in advance of installation which high level-protocols will be used.

Bridges form logically single networks. All interconnected network segments have the same network address. A bridge makes the movement of network devices within the network easy. There is no need to configure new network addresses for these devices.

When using transparent bridges, no network management is necessary. The bridges configure themselves to the topology automatically.

Transparent bridges learn about bridge and LAN failures and other topology changes quickly and automatically.

Some low-level protocols such as DEC's Local Area Transport (DECLAT) and NetBIOS cannot be routed because they contain no network layer information. These protocols must be bridged between segments.

Bridges are cheaper than routers because of the underlying simplicity of the architecture.

Bridges are simple to install. To use advanced bridging features, such as, custom filters, a minimal amount of configuration is required. In this case, an easy-to-use interface helps to facilitate such configuration.

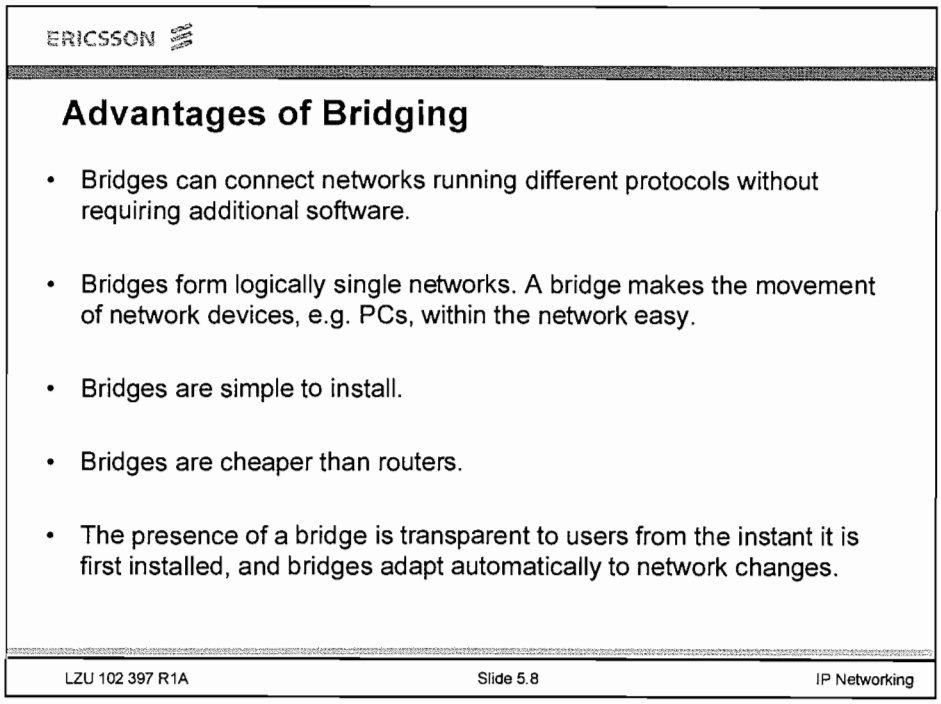


Figure 5-7.

Notes:

DISADVANTAGES OF BRIDGING

Bridges cannot load-share traffic over two paths to a single destination, because the STA ensures that one of these paths blocks all traffic. This is very expensive in the case of wide area links. For example, if a company purchased two separate 2 Mbit/s links to a remote site, only one of these could send traffic at any one time.

Bridges cannot prevent a broadcast storm. This may occur with certain broadcast protocols, which cause frames to be flooded out every port. If there is a malfunction or an incorrectly configured parameter on any network device, the level of traffic generated can be severe enough to crash the entire network.

Bridges do not provide significant support for fault isolation or other distributed management capabilities. Networks become harder to manage and maintain as their size and complexity increases. Bridges form a single logical network, often making fault isolation in very large bridged networks almost impossible.

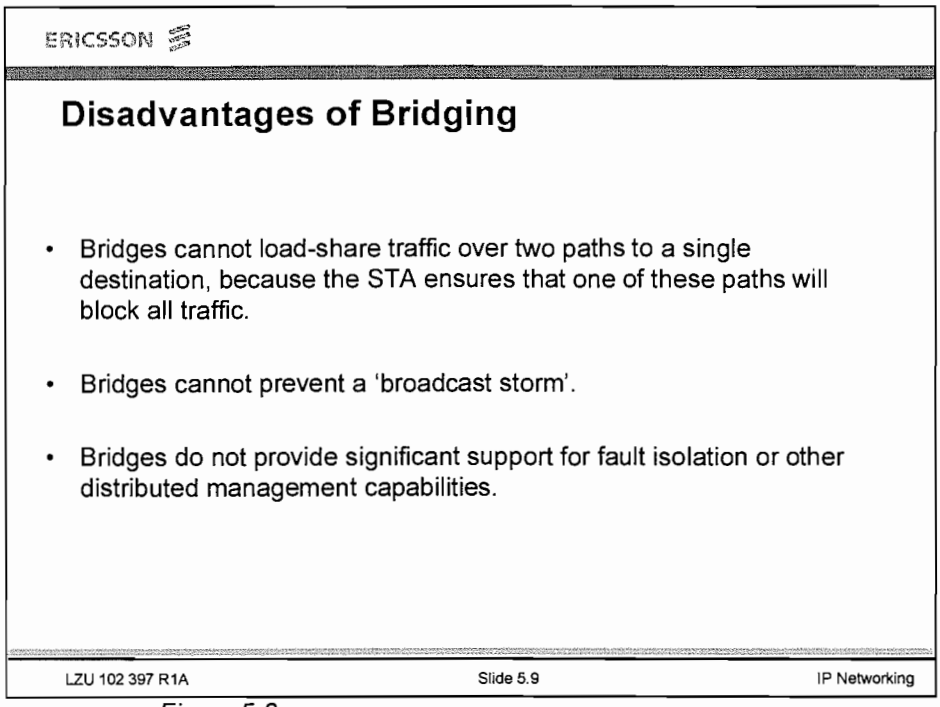


Figure 5-8.

Notes:

LAN SWITCHES

A LAN switch is a network device containing a high-speed backplane (> 1 Gbit/s) and typically support a large number of Ethernet connections. LAN Switches provide Gigabit Ethernet switching in a versatile, affordable platform that's simple to scale and manage. Switches are designed especially for organisations with diverse and evolving bandwidth requirements that want to lower their price per port and cost per megabit.

LAN Switches typically provide auto-sensing 10/100 Mbit Ethernet ports. For scalability, switches typically feature expansion slots that can accommodate optional Gigabit Ethernet LX, SX, or BASE-T modules.

Switches offer Layer 2 functionality, Layer 3 switching for IP networks, and advanced traffic prioritization and security capabilities to accommodate the ever growing requirements of core Gigabit backbones.

When a device sends a frame, it first arrives at a port on the switch. Each input port is buffered, so incoming frames are stored in the on-board RAM as they arrive. This design allows all input ports to receive and transmit frames at the same time, for parallel full-duplex operation.

If only a single device is connected to a port, each port is a separate collision domain, so collisions do not occur. It is possible to connect an Ethernet hub to a port on the switch, as both use standard Ethernet frames. Frames arriving at the switch from the hub are treated there like any other incoming frames; that is, they are switched to the correct output line over the high-speed backplane.

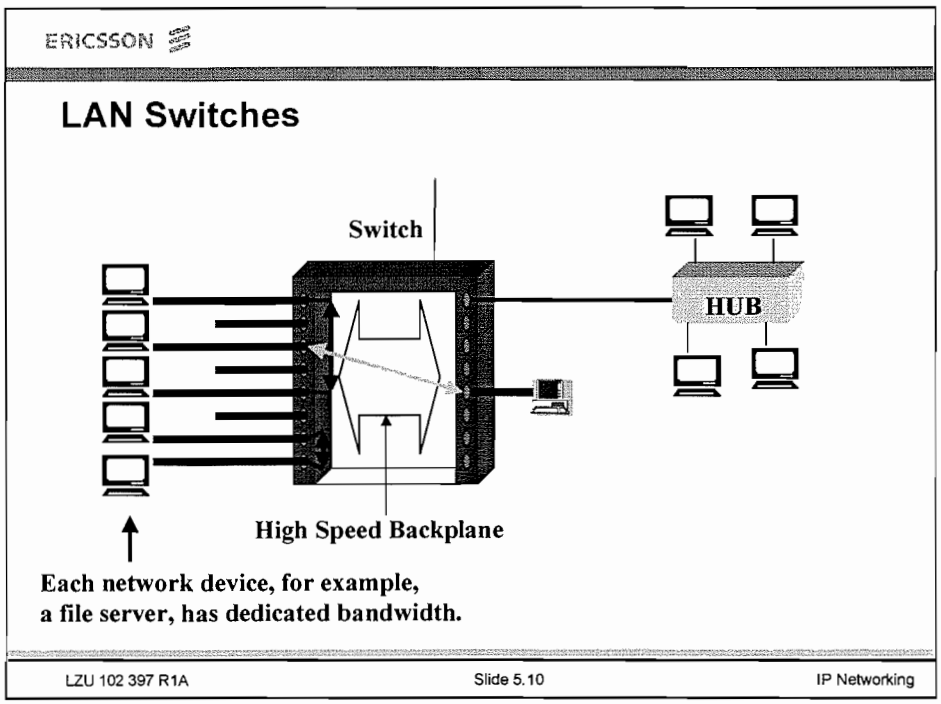


Figure 5-9.

Notes:

Intentionally Blank

6 Routing

After completing this chapter you will be able to:

- Understand how Routers operate
- Describe Distance Vector and Link State Algorithms
- Describe hierarchies in IP Routing
- Outline the advantages and disadvantages of Routing

Intentionally Blank

ROUTER OPERATION

Routers interconnect the various network segments making up the Internet. A router receives an IP packet on one of its interfaces, and forwards the packet out another of its interfaces (or possibly more than one if the packet is a multicast packet), in accordance with the contents of the IP header.

As the packet is forwarded hop by hop the packet's network-layer header, the IP header, remains relatively unchanged. However, the data link headers and physical transmission schemes may change radically at each hop in order to match the changing media types.

We will now examine what happens when a router receives a packet from one of its attached Ethernet segments. If the Ethernet type is set to 0800, indicating an IP packet, the Ethernet header is stripped from the packet, and the IP header is examined. Before discarding the Ethernet header, the router notes the length of the Ethernet packet and whether the packet has been multicast or broadcast on the Ethernet segment by checking a particular bit in the destination MAC address. In some cases routers will refuse to forward data link multicasts or broadcasts.

The router then verifies the contents of the IP header by checking the Version, Internet Header Length (IHL), Length, and Header Checksum fields. The version must be equal to 4. The IHL must be greater than or equal to the minimum IP header size (five 32-bit words). The length of the IP packet expressed in bytes must be also larger than the minimum header size. In addition, the router should check that the entire packet has been received, by checking the IP length against the size of the received Ethernet packet. Finally, to verify that none of the fields of the header have been corrupted, the 16-bit ones-complement checksum of the entire IP header is calculated and verified.

If any of these basic checks fail, the packet is deemed so malformed that it is discarded without even sending an error indication back to the packet's originator.

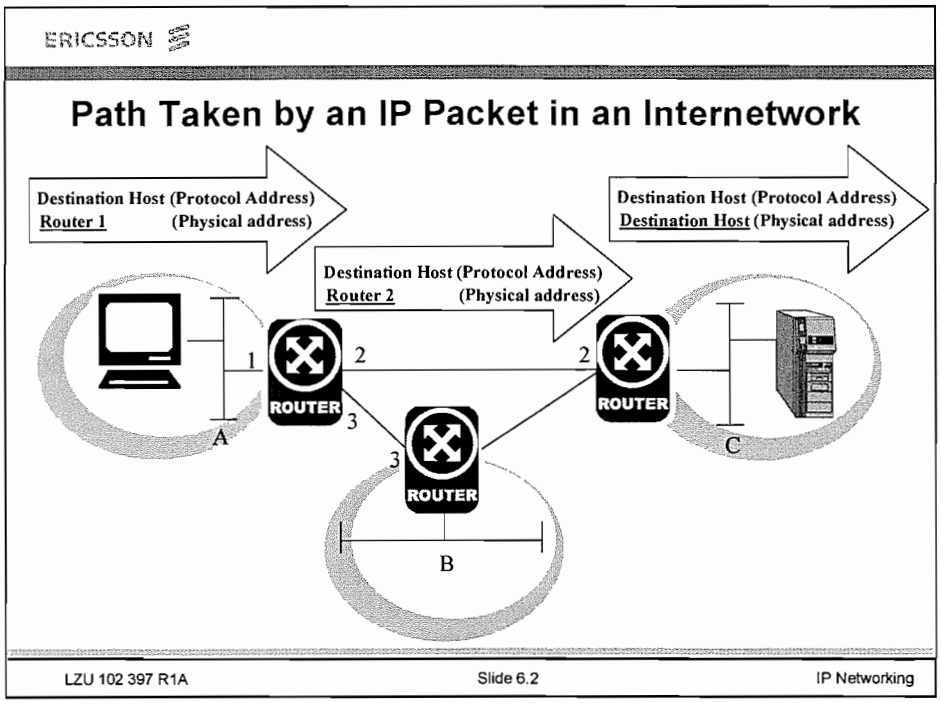


Figure 6-1.

Notes:

Router Operation (cont)

Next, the router verifies that the Time To Live (TTL) field is greater than 1.The purpose of the TTL field is to make sure that packets do not circulate forever when there are routing loops. Each router decrements the TTL field on the way to a destination. When the TTL field is decremented to 0, the packet is discarded, and an Internet Control Message Protocol (ICMP) TTL Exceeded message is sent back to the host. On decrementing the TTL, the router must adjust the packet's Header Checksum.

The router then looks at the destination IP address. The destination IP address is used as a key for the routing table lookup. The best matching routing table entry is returned, indicating whether or not to forward the packet. If the packet is to be forwarded, this entry also indicates the interface to forward the packet out of and the IP address of the next IP router.

If the packet is too large to be sent out on the outgoing interface in one piece, that is, its length is greater than the outgoing interface's Maximum Transmission Unit (MTU), the router attempts to split the packet into smaller pieces, called fragments. Fragmentation may affect performance adversely.

Hosts may wish to prevent fragmentation by setting the Don't Fragment (DF) bit in the Fragmentation field. In this case, the router drops the packet and sends an ICMP Destination Unreachable message back to the host. The host uses this message to calculate the minimum MTU along the packet's path, which is used to size future packets.

The router then prepares the appropriate data-link header for the outgoing interface. The IP address of the next hop is converted to a data-link address, usually using ARP or a variant of ARP, such as Inverse ARP. The router then sends the packet to the next hop, where the process is repeated.

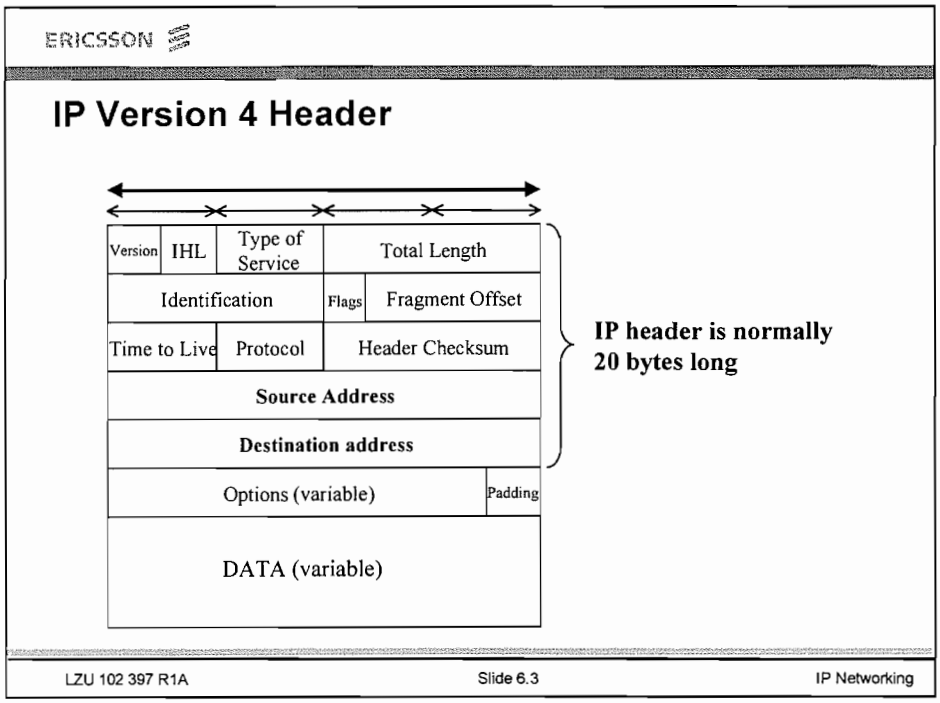


Figure 6-2.

Notes:

Finding the First Hop Router

We have described how a router forwards an IP packet. However, to start with, an IP packet sent from a host in one network to a destination in another network must find a router to send a packet to. There are two ways this is done in an Ethernet LAN. These are:

- Using a default gateway
- Using proxy ARP

USING DEFAULT GATEWAY

When a host sends a packet, it must determine the next hop. A host that has one network connection, such as an Ethernet interface, has an IP address assigned to it. The first test that the host performs is to determine whether the packet's destination address belongs to the same subnet. A logical AND is performed with the subnet mask and the destination IP address and compared to the result of a logical AND between the subnet mask and the host's own IP address. If the result is different, the destination is remote and the next hop's address is of a router on the path to this remote location. The host is configured with the IP address of the next hop router, that is, the default gateway.

The host must now find the hardware address of the default gateway. The host broadcasts an ARP request packet over the Ethernet. All stations receive this. The default gateway recognises the IP address and sends back an ARP reply. The hosts keep the result of the translation in their cache memories. If the host has to send more packets to the same destination it simply looks into the cache and copies the 48-bit hardware address without having to resort to ARP. Requests and responses are identified by the operation code (resp 1 and 2).

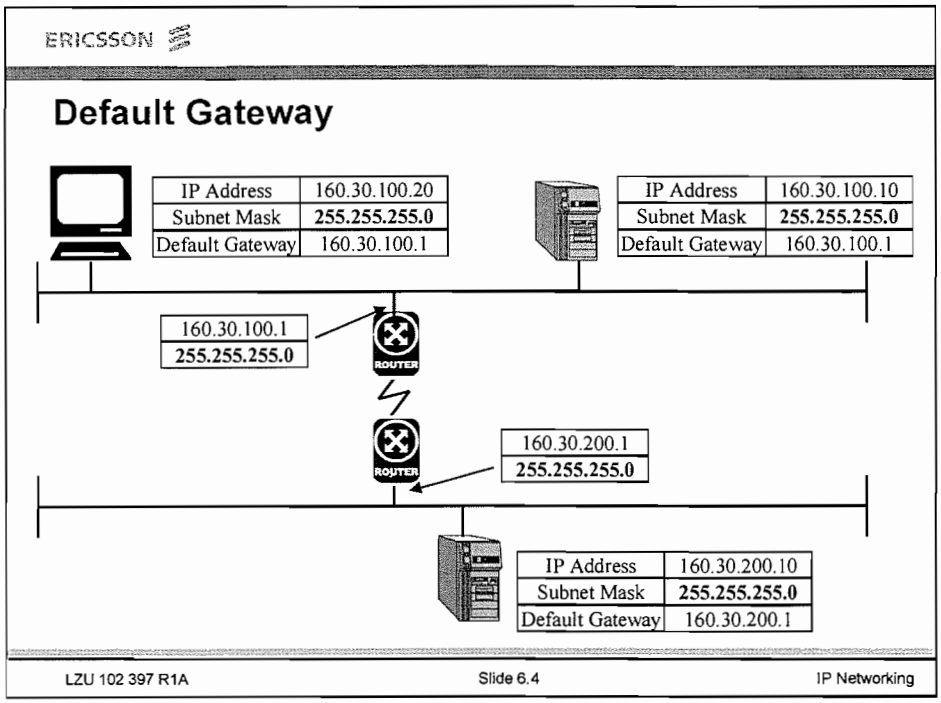


Figure 6-3.

Notes:

USING PROXY ARP

In the example in the following figure the PC and servers are configured with the class B default subnet mask, (255.255.0.0). The routers are configured with the customised mask (255.255.255.0).

If the PC wants to send a packet to the server on the remote network, it compares the destination's network ID and subnet, with its own network ID and subnet. The result implies that they are on the same network, so that the PC tries to send the packet directly using ARP.

Routers do not normally propagate broadcasts, so the actual ARP broadcast does not go beyond the senders network.

However, routers can run a protocol called Proxy ARP. When a router running Proxy ARP receives an ARP request, it reads the packet and applies the subnet mask for the sender's subnet to the requested destination IP address. This gives it the network ID that it compares with its routing tables in order to find a match.

In the example, the router determines that it knows a viable route to get the packet to the subnet of the destination. The router then replies to the ARP request exactly as if the router were itself the destination device. The only difference is that the hardware address returned in the ARP reply is the address of the router port connected to the source network. The source device and router now enter each other's IP/Hardware address pair in their ARP cache and the first data packet can be sent.

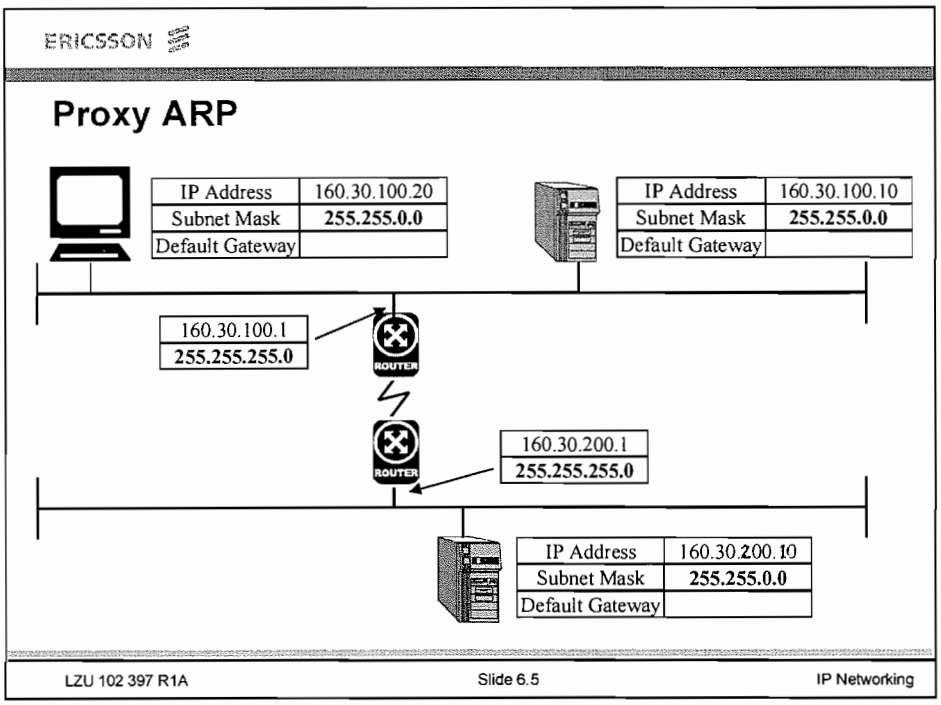


Figure 6-4.

Notes:

ROUTING TABLES

All TCP/IP routing protocols have ways of discovering the reachable IP address prefixes and, for each prefix, the next-hop router to use to forward data traffic to the prefix. As the network changes - leased lines fail, new leased lines are provisioned, routers crash, and so on - the routing protocols continually reevaluate prefix reachability and information about the next hop to use for each prefix. The process of finding the new next hop after the network changes is called convergence. Routing protocols that find the new next hop quickly, that is, protocols having a short convergence time, are preferred.

A router's routing table instructs the router how to forward packets. There is a separate routing table entry for each address prefix that the router knows. Entries in the routing table are also commonly known as routes. If a packet's IP destination falls into the range of addresses described by a particular routing table entry's prefix, we say that the entry is a match.

Many routers have a default route to external destinations in their routing table, that is, destinations that are not within the routing domain. The default route matches every destination, although it is overwritten by all the more specific prefixes.

There are two types of routing, which are:

- Dynamic routing
- Static routing

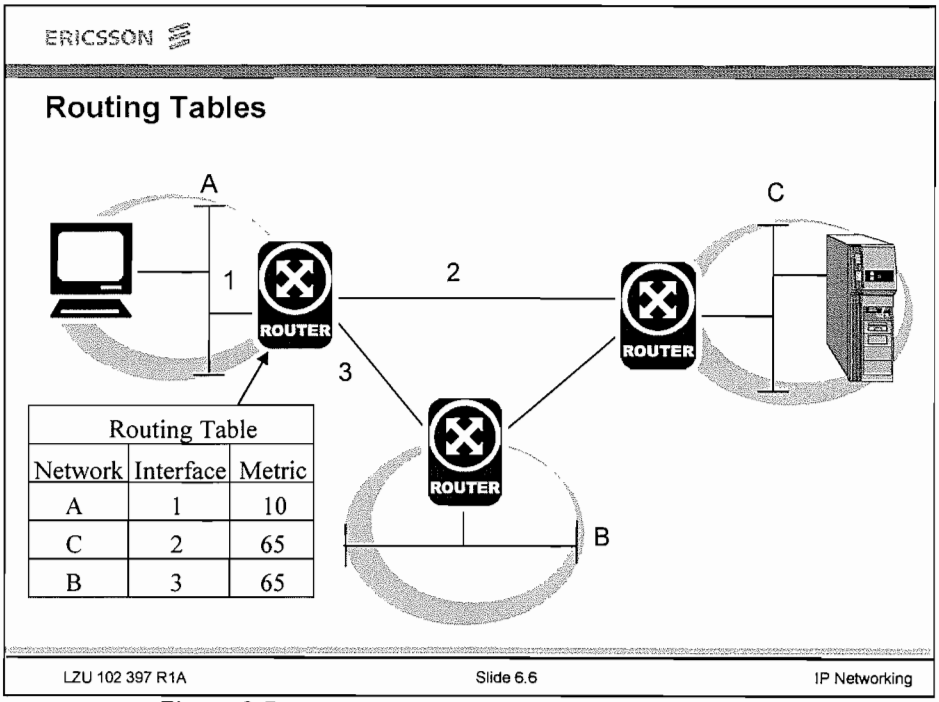


Figure 6-5.

Notes:

Static Routing

Static Routing table entries configured by a network operator is called static routing. For example, to install a route to subnet 192.168.10.0, a network operator may type the command:

ADD IP ROUTE ENTRY 192.168.10.0 255.255.255.0 192.168.12.1 1

Where 192.168.10.0 is the destination prefix (class C network), 255.255.255.0 is the network mask, and 192.168.12.1 is the IP address of the next hop router, and 1 is the metric associated with that path. One main disadvantage of static routing is if a link fails, an alternative link has to be configured manually by the network administrator.

DYNAMIC ROUTING

Dynamic routing adjusts in real time to network changes by analysing routing update messages. There are two main processes involved in dynamic routing.

- Information distribution, where each router sends and receives routing information within the internetwork using a routing protocol such as, BGP, OSPF and RIP.
- Route calculation, where each router calculates the best path to each destination using an algorithm and the information received using the routing protocols.

There are two types of algorithms used in routing. These are:

- Distance vector algorithms
- Link state algorithms

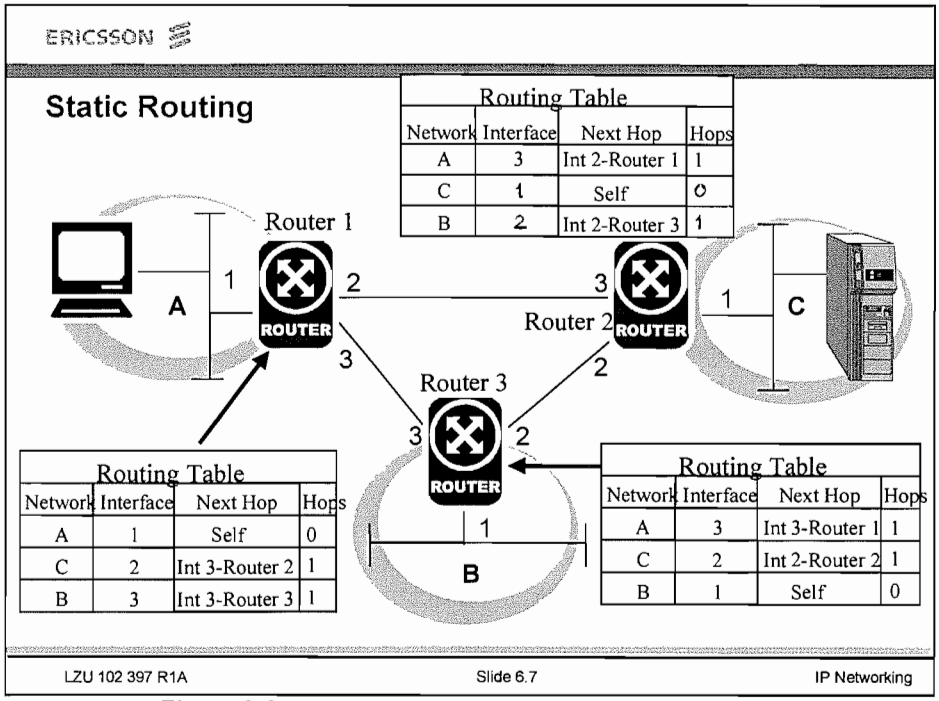


Figure 6-6.

Notes:

DISTANCE VECTOR PROTOCOLS

In a Distance Vector Protocol, the routers cooperate in performing a distributed computation. Distance Vector algorithms calculate the best path to each destination separately, usually trying to find a path that minimises a simple metric, such as the number of hop counts to the destination. Each route has its current best path to a destination and sends this information to its neighbours in routing updates. The router's neighbours also notify the router of their path choices. The router, seeing the paths being used by all of its neighbours, may find a better (that is, low cost) path to a particular destination through one of its neighbours. If so, the router updates its next hop and cost for the destination and notifies its neighbours of its new choice of route, and the procedure iterates. After some iterations, the choice of router stabilises, with each router having found the best path to the destination.

The main advantage of Distance Vector algorithms is their simplicity. The Internet's Routing Information Protocol (RIP) is a good example of a Distance Vector Protocol which uses the Bellmann Ford algorithm.

Link State Algorithms

Link State routing algorithms employ a replicated distributed database approach. Each router contributes pieces of information to this database by describing the router's local environment: the set of active links to local IP networks and neighbouring routers, with each link assigned a cost. Instead of advertising a list of distances to each known destination, a router running linkstate algorithm advertises the states of its local network links. These link state advertisements are then distributed to all routers. The end result is that all routers obtain the same database of collected advertisements, together describing the current map of the network. From the network map, each router runs a shortest-path calculation, typically the Dijkstra algorithm. The shortest path in the network is assigned as the sum of the costs of the links comprising the path.

Link-state algorithms are considered to have good convergence properties. When the network changes, new routes are found quickly and with a minimum of routing protocol overhead. Link-state routing protocols are more complicated to specify than are Distance Vector Protocols, as you can tell by comparing the size of the OSPF and RIP specifications.

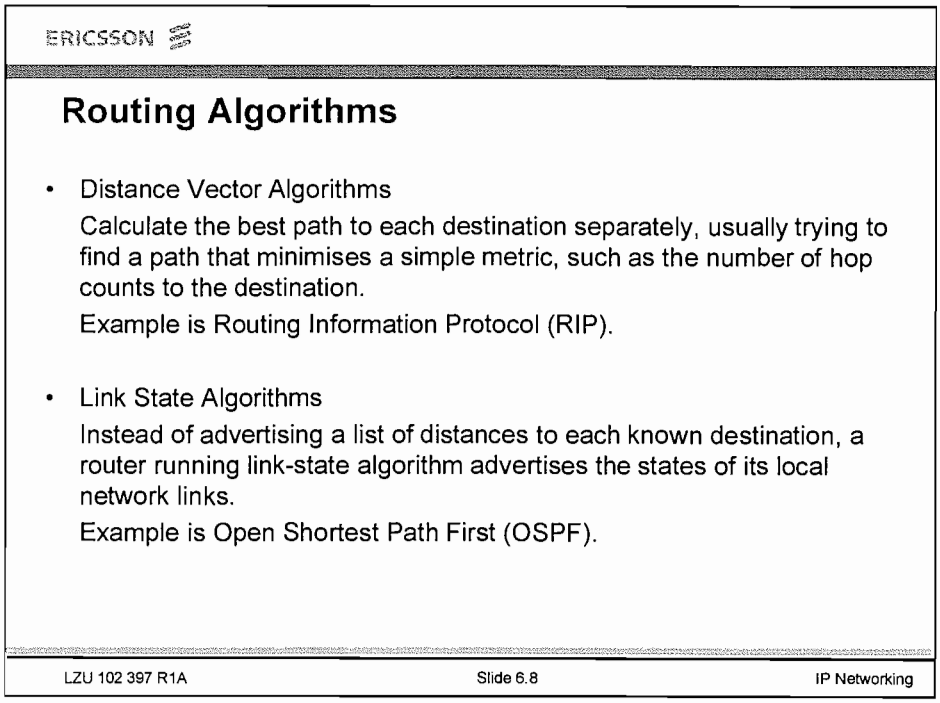


Figure 6-7.

Notes:

ROUTING METRICS

Metrics are used by routing algorithms to select the best route. Sophisticated routing algorithms can use a combination of the following metrics:

- Path length is the sum of the interface costs associated with each network link. Hop count specifies the number of passes through internetworking devices (such as routers) that a packet must take from a source to a destination.
- Reliability is usually assigned to network links by network administrators. The values assigned are based on how frequently the network link goes down and how long it typically takes to be repaired.
- Delay refers to the length of time it takes to move a packet from source to destination through an internetwork. It is dependent on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, network congestion on all intermediate network links, and the physical distance to be travelled.
- Bandwidth refers to the available traffic capacity of a link.
- Load refers to the degree to which a network resource (such as a router) is busy, for example, its CPU utilisation and the number of packets processed per second.
- Communications cost is the actual financial cost associated with a particular route. A network administrator may configure routers so that traffic uses a slower link, if it is cheaper to do so.

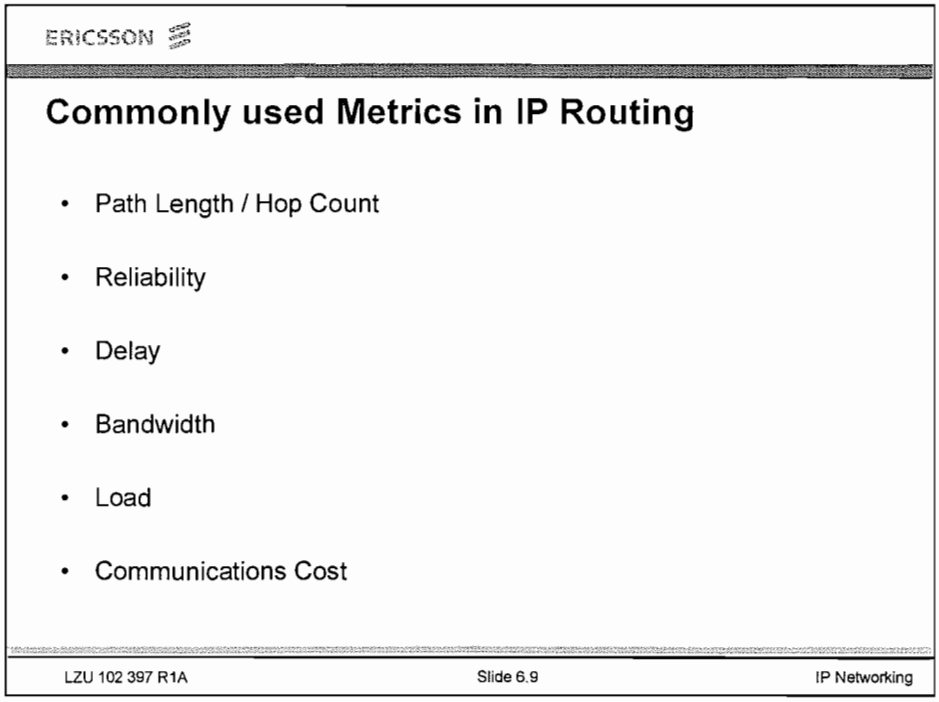


Figure 6-8.

Notes:

DISTANCE VECTOR ALGORITHM

The following figure shows a small network with several nodes. Each node knows about its own links. Initially the routing tables have a single entry for the node itself.

Node A abstracts this information in a distance vector, that is A=0. It broadcasts this distance vector to all its neighbours. Therefore B and D receive this information and are able to enlarge their knowledge.

The vector A=0 received from node A is incremented by 1 to account for the cost of getting from Node B or Node A.

Node B can now prepare its own distance vector (B=0, A=1) and send it on the local links, that is, 1, 2 and 4.

Similarly, D transmits its own distance vector (D=0, A=1) on links 3 and 6. The message from B is received by A, C and E.

A calculates the distances to B=1, to D=1 and to A=2, via B. It observes that the value for A is larger than that of the local entry and thus ignores it. The routing table in A eventually looks like:

DIJKSTRA ALGORITHM

The Dijkstra algorithm involves using the information in the link-state database to calculate the routing tables.

The Dijkstra algorithm is a simple algorithm that efficiently calculates all the shortest paths to all destinations at once. The algorithm incrementally calculates a tree of shortest paths. It begins with the calculating router adding itself to the tree. All of the router's neighbours are then added to a candidate list, with costs equal to costs of the links from the router to the neighbours. The router on the candidate list with the smallest cost is then added to the shortest path tree, and that router's neighbours are then examined for inclusion in the candidate list.

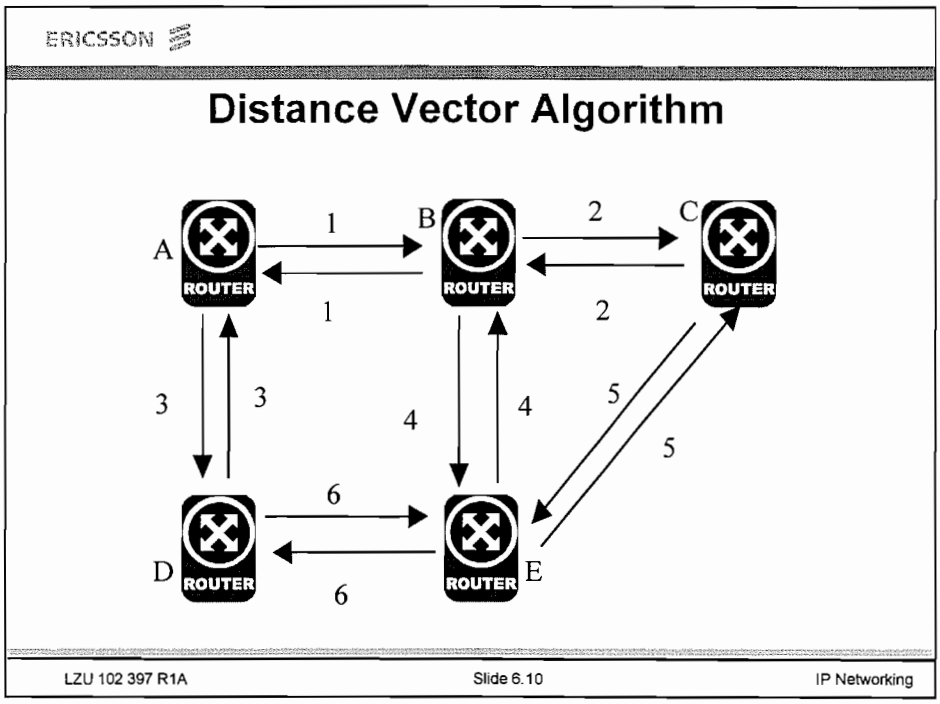


Figure 6-9.

From A to	Link	Cost
A	Local	0
B	1	1
C	1	2
D	3	1
E	1	2

Table 6-1.

IP ROUTING PROTOCOL HIERARCHIES

In the early 1980,s, the Internet was a single network. All routers, which were then called "gateways", shared routing information through the same gateway to gateway (GGP) protocol. The routing tables included entries for all IP networks in the Internet.

This configuration caused a number of problems. The routing overhead increased with the number of connected routers. The size of routing table increased with the number of connected networks. The frequency of the routing updates also increased. As the number of routers and links increased the more unstable the network became and hence the more frequent the number of routing updates. As the number of routers increased, so too did the number of different types of routers. Different machines from different manufacturers were increasingly being used. All these machines used their own specific implementation of GGP, which made maintenance and fault isolation almost impossible.

It was decided to split the Internet into a set of Autonomous Systems (AS).

Each AS comprised of a set of routers and networks under the same administration. One AS was formed from ARPANET and Satnet routers. This formed the core and played a "backbone role". All the other autonomous systems would connect to the backbone using a router called an exterior gateway.

All routers within the same AS are interconnected. These routers exchange routing information. This is normally done by selecting a single routing protocol and running it between all the routers. In 1982 terminology, routers inside an AS were called "interior gateways" and the protocol was an "Interior Gateway Protocol" (IGP). Examples of IGP in use today are RIP, OSPF and IGRP. These routers can discover information only about the internal networks to which they are directly connected. They must get information about exterior networks through a dialogue with exterior gateways, which are entry points into adjacent autonomous systems.

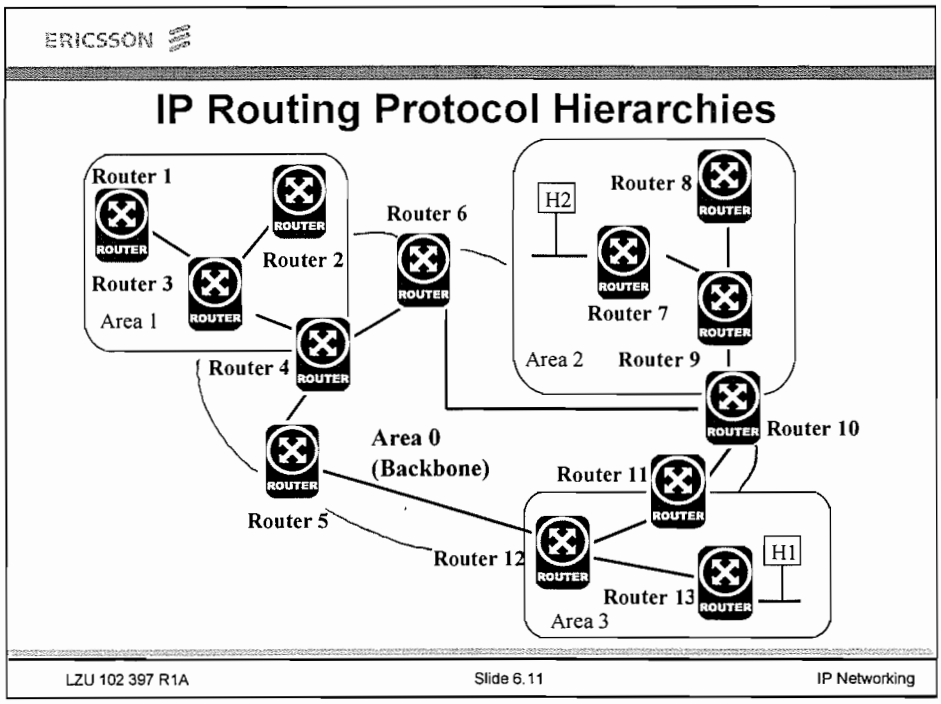


Figure 6-10.

Notes:

IP Routing Protocol Hierarchies(cont)

The protocol used between autonomous systems is called Exterior Gateway Protocol (EGP). EGP organises the exchange of information between two adjacent autonomous systems. This involves three separate procedures:

- Neighbour acquisition
- Neighbour reachability
- Network reachability

Neighbour acquisition

Before exchanging any information and indeed before using any routing information, the adjacent routers must agree to become neighbours for EGP. The neighbour acquisition procedure is a simple "two-way handshake". The router that wishes to become a neighbour sends a "neighbour acquisition request" to its partner, which will reply with an acquisition reply", The partner may also refuse to become a neighbour and reply with a "refusal" message.

When a request/reply exchange has been successfully performed, the routers become neighbours.

Neighbour Reachability

The purpose of the neighbour reachability procedure is to check that the link to the neighbour is still operational. The router that wants to check reachability sends a "hello" message at regular intervals. The neighbour sends a "I heard you" message in response.

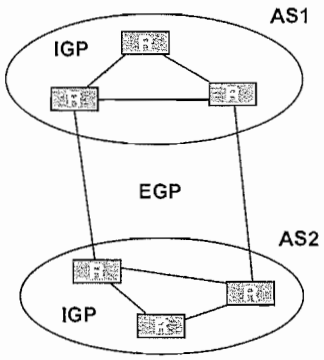
Network Reachability

The purpose of the network reachability procedure is to exchange the list of networks that can be reached through each neighbour. The procedure is based on "polling"- each neighbour, at regular intervals, polls its partner for a list.

ERICSSON

Choosing Routing Protocols

- Interior Gateway Protocols
  - Distance vector, Link state
  - RIP, OSPF or IS-IS
- Exterior Gateway Protocols
  - Default routes, Routing policy
  - BGP or IDRP



LZU 102 397 R1A Slide 6.15 IP Networking

Figure 6-11.

Notes:

ADVANTAGES OF ROUTERS

- Routers are generally more flexible than bridges. They can differentiate between different paths on the basis of factors such as cost, line speed, and line delay.
- Routers can be configured for equal-cost load splitting. This means that they can take advantage of all communication paths simultaneously, and purchased bandwidth is not placed in stand-by mode.
- Network devices recognise when they are communicating through a router. If network congestion occurs, routers use a mechanism called source quench, which indicates to network devices that they must slow down.
- Routers provide the network administrator with more control over resources on the network. Because each segment has a different address, it is easier for the administrator to track what is running on the network and where.
- Routers provide a protective firewall between network segments. This protects against broadcast storms and prevents incidents that occur on one segment from affecting another.

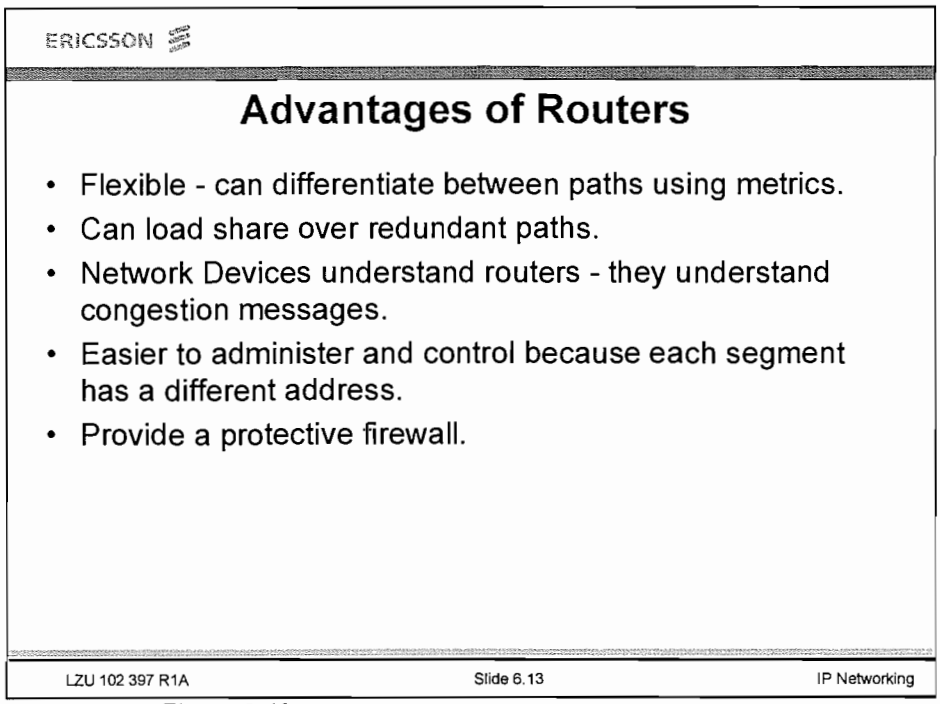


Figure 6-12.

Notes:

DISADVANTAGES OF ROUTERS

- Because routers are protocol-dependent devices, they require software for each protocol that they run. Software for each individual protocol must be separately installed.
- Routers make movement of network devices between network segments more difficult. Since each segment has a different network address, moving a device from one segment to another requires that the network administrator assign a new network address to the relocated network device.
- Some low-level protocols, for example, DEC LAT and NetBIOS, cannot be routed because they do not contain a network layer address. These protocols must be bridged.

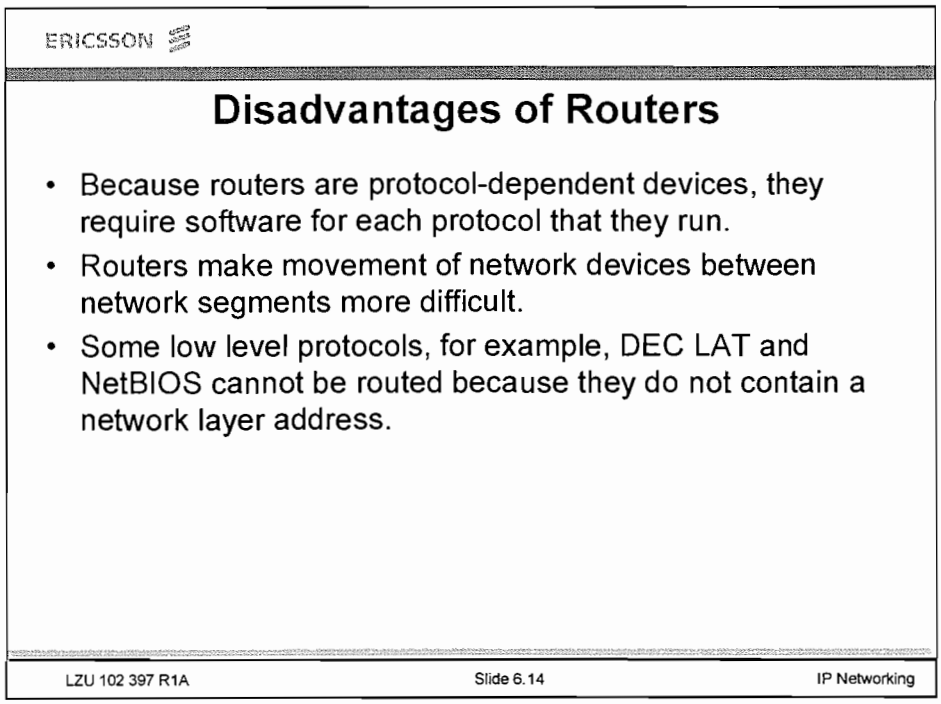


Figure 6-13.

Intentionally Blank

7 Routing Information Protocol (RIP)

Intentionally Blank

ROUTING INFORMATION PROTOCOL (RIP)

One interior gateway protocol in today's Internet is Routing Information Protocol (RIP). RIP is a very simple protocol of the distance vector family. The RIP protocol is based on the Bellman-Ford algorithm. RIP allows hosts and gateways to exchange information for computing routes through an IP-based network.

RIP was initially designed as a component of the networking code for the BSD (Berkley System Design) release of UNIX, incorporated into a program called "routed", which is the short for "route management daemon". It is an extremely simple protocol requiring minimal configuration. RIP was built and adopted widely before a formal standard was written. Most implementations were derived from the Berkley code. RIP was documented in RFC-1058 in June 1988 by Charles Hedrick which made it possible for vendors to ensure interoperability.

A router running RIP broadcasts a message every 30 seconds. The message contains information taken from the router's current routing database. Each message consists of pairs, each pair contains an IP network address and an integer distance to that network.

The addresses used in RIP are 32-bit Internet addresses. An entry in the routing table can represent a host, a network or a subnet.

By default, RIP uses a very simple metric: the distance is the number of hops to be used to reach the destination. This is normally called the hop count. This distance is expressed as an integer varying between 1 and 15; the value 16 denotes infinity.

RIP supports both non-broadcast and broadcast networks. RIP packets are carried over User Data Protocol (UDP) and IP. The RIP processes uses UDP port number 520 for emission and reception. Packets are normally sent every 30 seconds, or more frequently in the case of triggered updates. If a route is not refreshed within 180 seconds, the distance is set to infinity and the entry is removed from the table later.

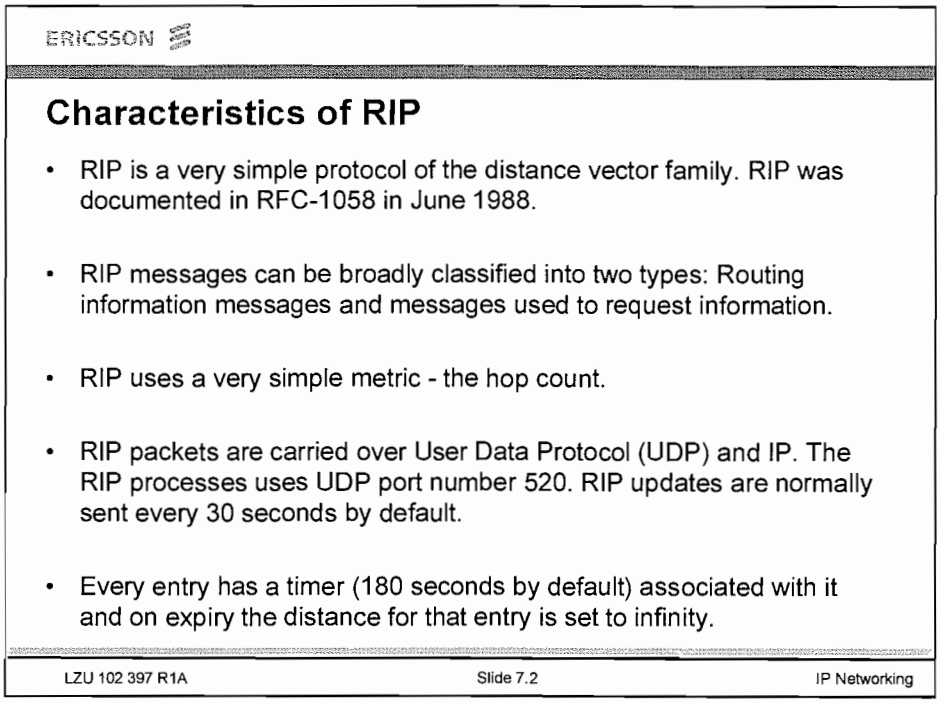


Figure 7-1.

Notes:

THE RIP PROTOCOL

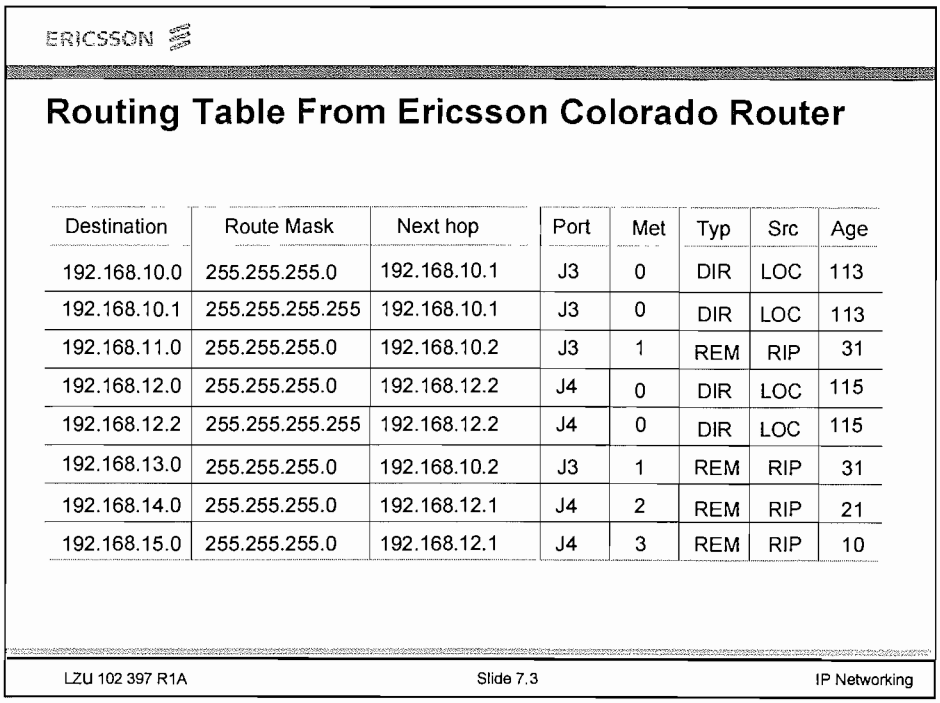
RIP messages can be broadly classified into two types; routing information messages and messages used to request information. Both use the same format, a fixed header followed by an optional list of network and distance pairs.

The RIP messages start with a 32-bit command and a version identifier, followed by a set of address and metric pairs, each of which is carried over 20 bytes.

Each host that implements RIP is assumed to have a routing table. This table has one entry for every destination that is reachable through the system described by RIP. Each entry contains at least the following information:

- The IP address of the destination.
- A metric, which represents the total cost of getting a datagram from the host to that destination. This metric is the sum of the costs associated with the networks that would be traversed in getting to the destination.
- The IP address of the next gateway along the path to the destination. If the destination is on one of the directly connected networks, this item is not needed.
- A flag to indicate that information about the route has changed recently. This will be referred to as the "route change flag."
- Various timers associated with the route.

RIP messages do not contain an explicit 'length' field. Instead, RIP assumes that the underlying delivery mechanism will tell the receiver the length of an incoming message. RIP relies on UDP to tell the receiver the message length. The address format is not limited for use with TCP/IP. It can be used with multiple network protocol suites. Each network address reported by RIP can have an address of up to 14 octets.



Destination	Route Mask	Next hop	Port	Met	Typ	Src	Age
192.168.10.0	255.255.255.0	192.168.10.1	J3	0	DIR	LOC	113
192.168.10.1	255.255.255.255	192.168.10.1	J3	0	DIR	LOC	113
192.168.11.0	255.255.255.0	192.168.10.2	J3	1	REM	RIP	31
192.168.12.0	255.255.255.0	192.168.12.2	J4	0	DIR	LOC	115
192.168.12.2	255.255.255.255	192.168.12.2	J4	0	DIR	LOC	115
192.168.13.0	255.255.255.0	192.168.10.2	J3	1	REM	RIP	31
192.168.14.0	255.255.255.0	192.168.12.1	J4	2	REM	RIP	21
192.168.15.0	255.255.255.0	192.168.12.1	J4	3	REM	RIP	10

Figure 7-2.

Notes:

The RIP Protocol (Contd)

Every datagram contains a command, a version number, and possible arguments. The command field is used to specify the purpose of the datagram. The commands for version 1 and 2 are:

Request: A request for the responding system to send all or part of its routing table.

Response: A message containing all or part of the sender's routing table. This message may be sent in response to a request or poll, or it may be an update message generated by the sender.

For request and response, the rest of the datagram contains a list of destinations, with information about each. Each entry in this list contains a destination network or host, and the metric for it.

The packet format is intended to allow RIP to carry routing information for several different protocols. Thus, each entry has an address family identifier to indicate what type of address is specified in that entry.

The address family identifier for IP is 2. The metric field must contain a value between 1 and 15 inclusive, specifying the current metric for the destination, or the value 16, which indicates that the destination is not reachable.

The maximum datagram size is 512 octets. This includes only the portions of the datagram described above. It does not count the IP or UDP headers.

Request is used to ask for a response containing all or part of the host's routing table. Normally, requests are sent as broadcasts, from a UDP source port of 520 and the request in IP networks always requests the contents of an entire routing table to be sent.

When a router receives a new RIP update it performs the following tasks. The router examines the entries one by one. It performs a set of validation checks to see, for example;

Incorrect entries are ignored. If the metric is not equal to infinity it is incremented by 1 to account for the last hop. The routing table is searched for the entry for that corresponding destination, and the distance vector processing is performed as follows:

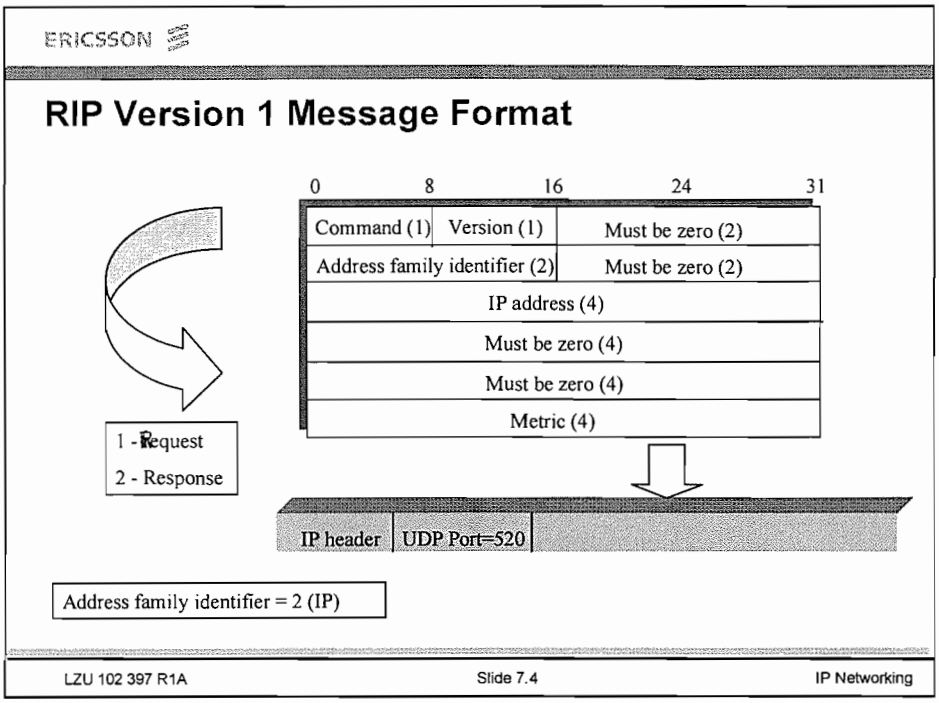


Figure 7-3.

Notes:

The Input Process

If the entry is not present, it is added, initializing the metric to the received value and a timer for the entry is started.

- Update the metric by adding the cost of the network on which the message arrived. If the result is greater than 16, use 16.
- If the entry was present with a larger metric, the metric is updated and next hop fields and the timer for the entry is restarted.
- If the new metric is the same as the old one, the router will ignore the entry.

The maximum message size chosen is 512 bytes, which allows for up to 25 entries per message. If there are more than 25 entries to report, RIP will send multiple packets.

The Output Process

A response is sent to the host at the opposite end of each connected point-to-point link, and a response is broadcast on all connected networks that support broadcasting. Thus, one response is prepared for each directly connected network and sent to the corresponding (destination or broadcast) address. In most cases, this reaches all neighbouring gateways.

To fill in the entries, the router goes down all the routes in the internal routing table. Recall that the maximum datagram size is 512 bytes. When there is no more space in the datagram, the current message is sent and a new one is started.

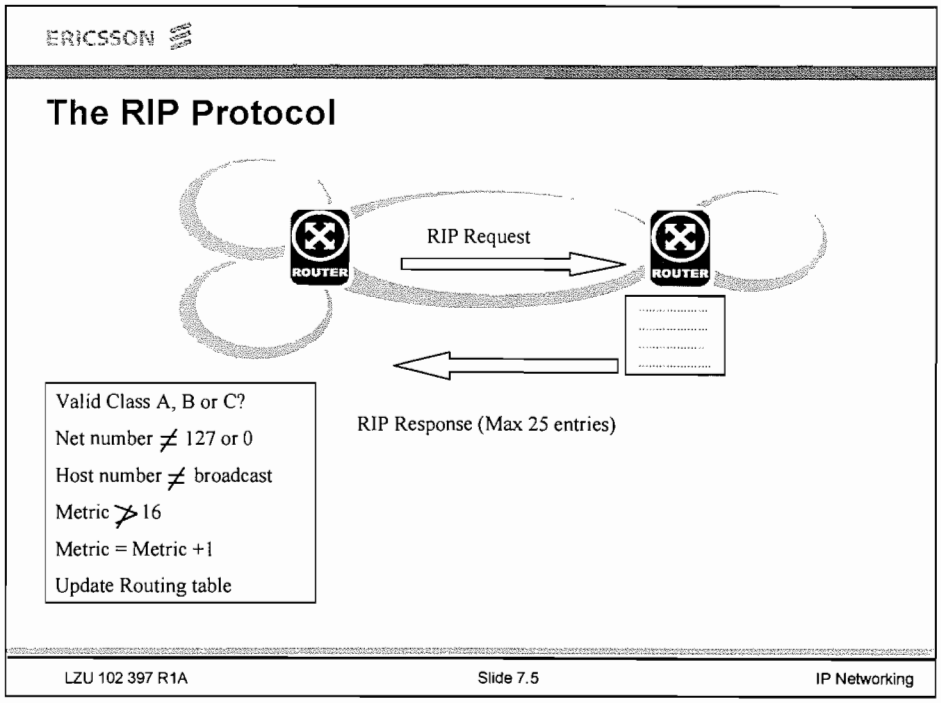


Figure 7-4.

Notes:

RIP NEIGHBOURS

RIP neighbours are defined as other RIP routers that are no more than one routing hop away (that is, a router at the other end of a point-to-point link). In most cases, RIP dynamically determines RIP neighbours (and thereby obtains routing information), over most media types. This eases the burden of manual configuration. However, there are some circumstances where additional manual configuration of RIP neighbours is mandatory or desirable.

Broadcast Network Interfaces

On broadcast networks, RIP, by default, behaves promiscuously. This means that RIP accepts all updates from any RIP router on the attached broadcast network. An example of a broadcast network is Ethernet. RIP updates are sent as subnet directed broadcasts, multicasts or unicasts, depending on how RIP is configured. Once RIP neighbours have been configured on an interface, only received updates from those listed neighbours are accepted.

Non-Broadcast Multi-Access (NBMA) Network Interfaces

On NBMA network interfaces, RIP requires at least one neighbour to be defined before the two routers across the NBMA network will exchange routing information. Examples of NBMA networks are Frame-Relay, X.25 and ATM.

Multicasting

In order to reduce unnecessary load on those hosts which are not listening to RIP-2 messages, an IP multicast address will be used for periodic broadcasts. The IP multicast address is 224.0.0.9. Note that IGMP is not needed since these are inter-router messages, which are not forwarded. On NBMA networks, unicast addressing may be used. However, if a response addressed to the RIP-2 multicast address is received, it should be accepted. In order to maintain backward compatibility, the use of the multicast address will be configurable. If multicasting is used, it should be used on all interfaces that support it.

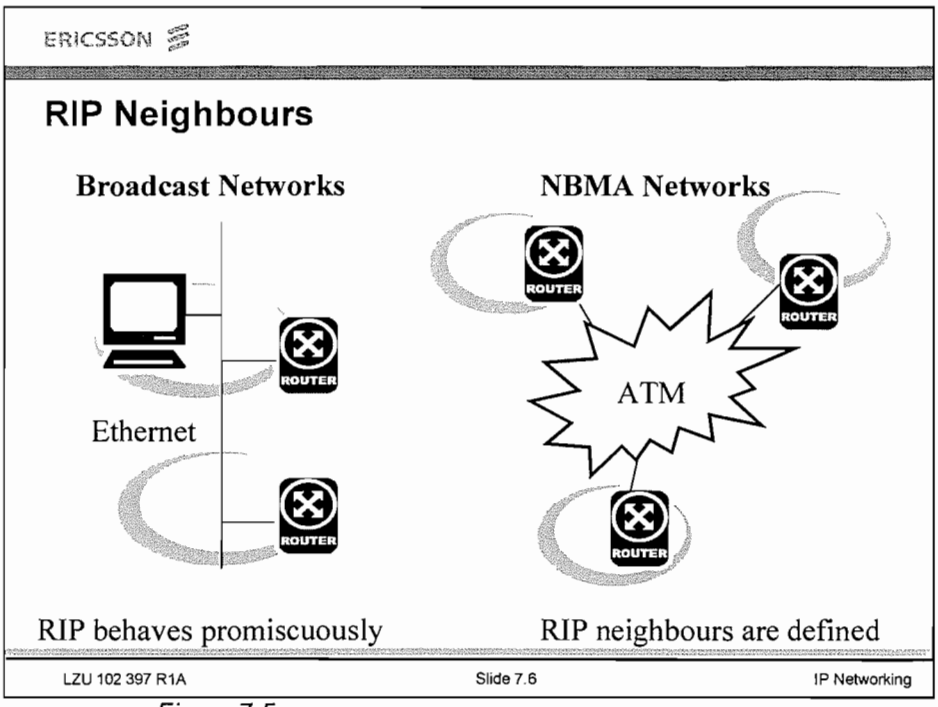


Figure 7-5.

Notes:

RIP VERSION 2

The RIP-1 format contains a number of 'must be zero' fields. These fields are redefined in RIP version 2. In RIP version 2, the subnet mask enables better subnet routing. The route tag is used to flag external routes and is used by EGP or BGP.

RIP-2 packets are sent with the version number set to 2. RIP-2 routers can interwork with RIP-1 routers. When receiving a packet with a version number larger than 1, RIP-1 routers simply ignore an entry where a 'must be zero' field has a non-zero value.

RIP-2 allows routing on the subnet outside of the network by passing mask information in parallel with the address.

RIP version 2 supports Classless Inter Domain Routing (CIDR). It also has enhancements that allow the network administrator to configure routing policy used by RIP.

The Route Tag (RT) field is an attribute assigned to a route which must be preserved and readvertised with a route. The intended use of the Route Tag is to provide a method of separating "internal" RIP routes (routes for networks within the RIP routing domain) from "external" RIP routes, which may have been imported from an EGP or another IGP. Routers supporting protocols other than RIP should be configurable to allow the Route Tag to be configured for routes imported from different sources. For example, routes imported from EGP or BGP should be able to have their Route Tag either set to an arbitrary value, or at least to the number of the Autonomous System from which the routes were learned.

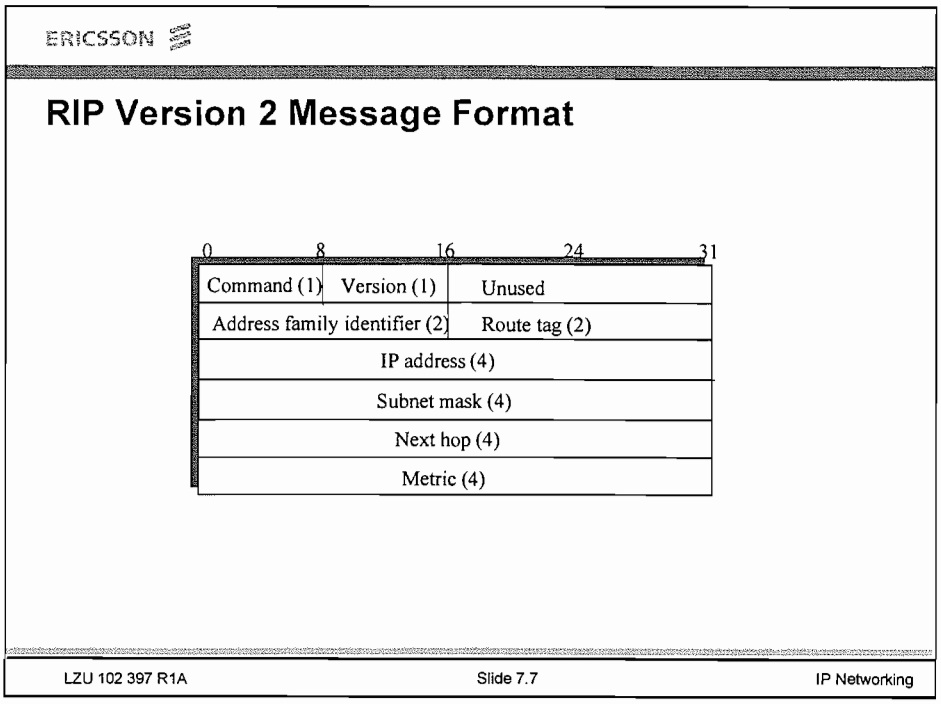


Figure 7-6.

Notes:

SLOW CONVERGENCE

The slow convergence problem can make routers wrongly believe they have a connection to a network, after the connection has failed. In the event of a failure, a router stops advertising a route and the protocol must depend on the timeout mechanism before it considers the route unreachable. Once the timeout occurs, the router finds an alternative route and starts propagating that information.

In the example in the following figure, Router 1 is aware that its link to network A has failed. However, Router 2 continues to broadcast RIP messages, stating that it can reach network A in 2 hops. Router 1 then assumes it can reach network A via Router 2 and changes its routing table to show the new route. Both Router 1 and Router 2 will continue to exchange RIP messages, increasing the hop count to network A each time, until the count reaches 15 and it is assumed to be unreachable.

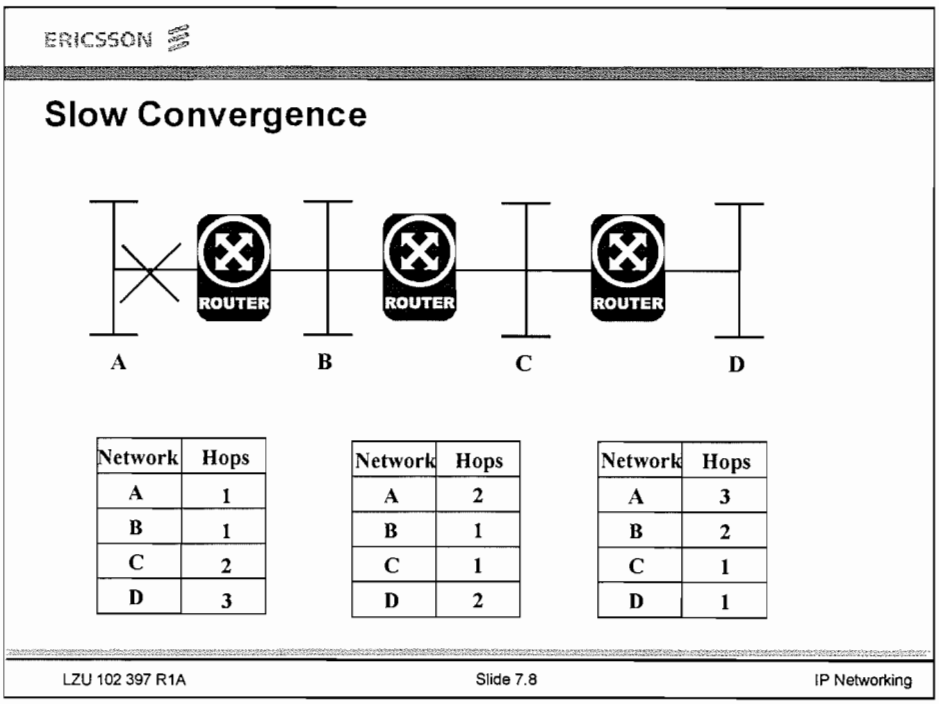


Figure 7-7.

Notes:

ROUTING LOOPS

RIP is unable to detect loops and uses a hop count of 15 to denote infinity. This means that packets could circulate around a loop until the hop count reached 15. When the hop count exceeds 15, the route is marked unreachable.

This is illustrated in the following diagram. We shall concentrate on the routes from various network nodes to node C. In a stable condition, after a successful cold start, we would have the following routes:

From	Link	Cost
A to C	1	2
B to C	2	1
C to C	Local	0
D to C	3	3
E to C	4	2

Table 7-1

Suppose that link number 2 breaks. This failure is immediately noticed by B, which updates the 'distance to C' to infinity. The situation is now:

From	Link	Cost
A to C	1	2
B to C	2	Inf
C to C	Local	0
D to C	3	3
E to C	4	2

Table 7-2.

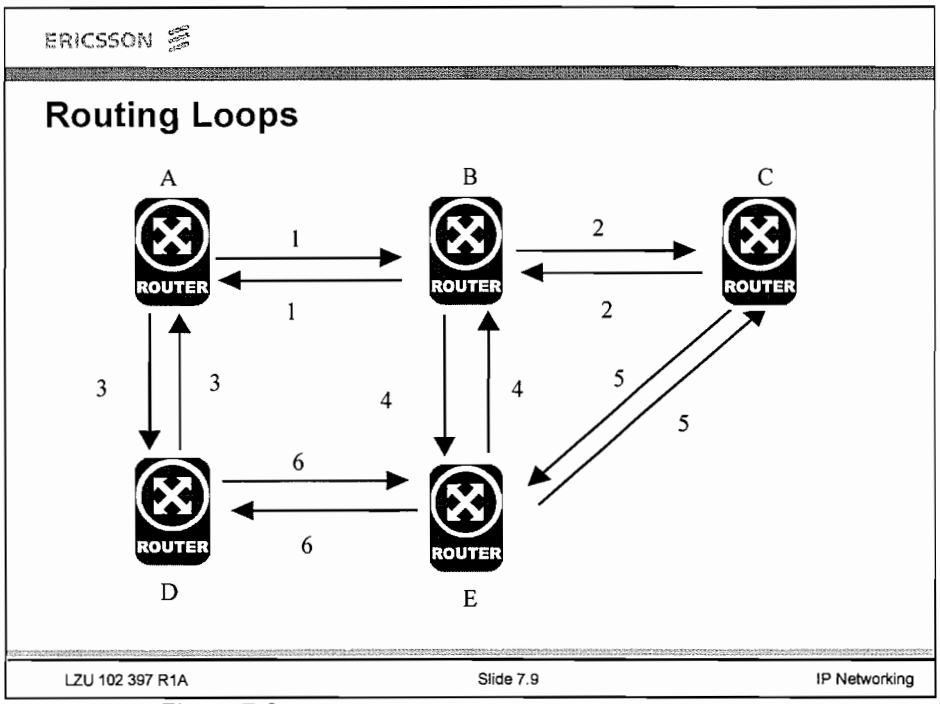


Figure 7-8.

Notes:

Routing Loops (Cont)

We assume that just before B has time to advertise its distance vector to its neighbours, A sends its own distance vector to B and D. B adds the cost 1 to the distance 2 advertised by A for C, and notice that the cost 3 is lower than the infinite cost present in the tables. It updates its own table to say that C is now reachable through link 1 at a cost of 3 and advertises this new distance to its neighbours A and E. The situation is now:

From	Link	Cost
A to C	1	4
B to C	1	3
C to C	Local	0
D to C	3	3
E to C	4	4

Table 7-3.

Note that the routing table now includes a loop, packets bound for C reach B and then bounce back and forth between A and B until their 'time to live' expires.

To prevent instabilities, the RIP algorithm must use a low value for the maximum possible distance (RIP uses 16). Thus, for internets in which legitimate hop counts approach 16, managers must divide the internet into two sections or use an alternative routing protocol.

Choosing a small infinity (16) helps limit slow convergence, but does not eliminate it.

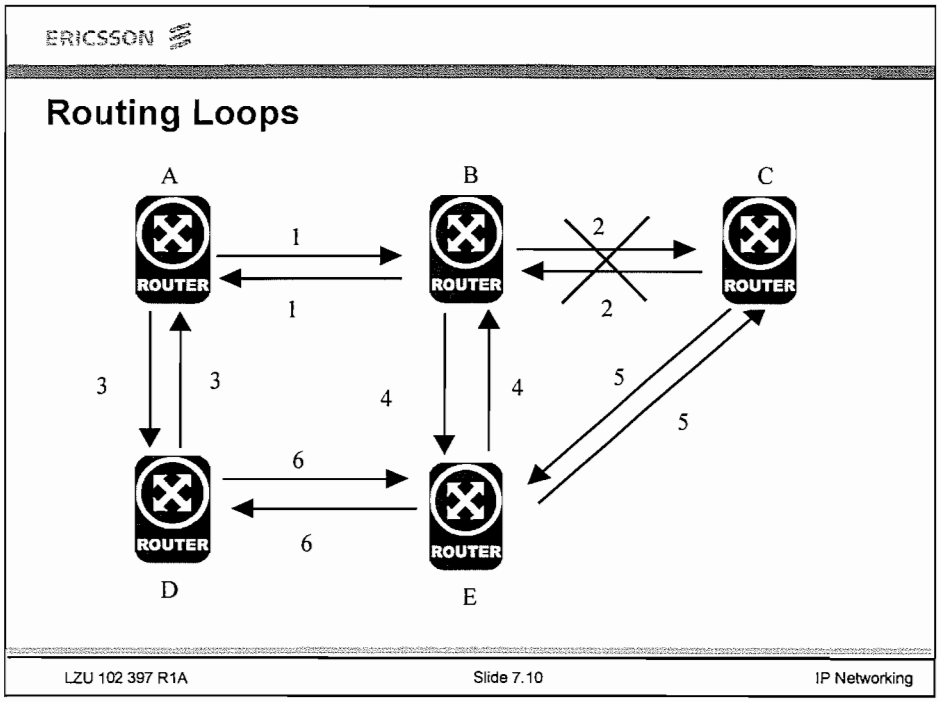


Figure 7-9.

Notes:

SPLIT HORIZON

Special techniques have been investigated to try and minimise the effect of loops. Split Horizon is based on a very simple precaution: if Node A is routing packets bound for destination X through node B, it makes no sense for B to try to reach X through A.

Split horizon comes in two variations.

In the first variation a router records the interface over which it received a particular route and does not propagate the information about that route back over the same interface. "Split horizon" is a scheme for avoiding problems caused by including routes in updates sent to the router from which they were learned.

SPLIT HORIZON WITH POISON REVERSE

The form known as 'split horizon with poison reverse' is more aggressive. Split horizon with poisoned reverse includes such routes in updates, but sets the corresponding distance to infinity if the destination is routed on the link. This immediately kills two-hop loops.

However, poisoned reverse does have the disadvantage in that it increases the size of the routing messages. Considering a large network with routers connected to a backbone. In this case if split horizon with poison reverse is used, the router must mention all routes that it learns from the backbone, with metrics of 16. This can result in large update messages, almost all of whose entries indicate unreachable networks.

In a static network, advertising reverse routes with a metric of 16 provides no useful additional information. If there are many routers on one broadcast network, these extra entries can use significant bandwidth. The reason they are there is to improve dynamic behavior. When topology changes, mentioning routes that should not go through the router as well as those that should can will speed up convergence.

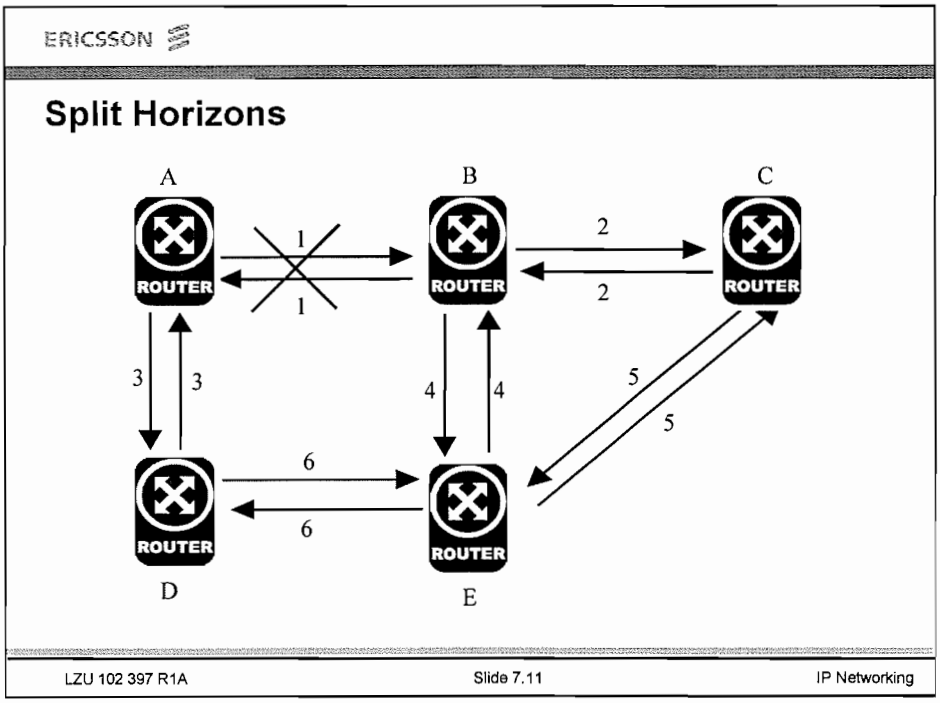


Figure 7-10.

Notes:

TRIGGERED UPDATES

Split horizon with poisoned reverse will prevent any routing loops that involve only two routers. It is still possible to end up with patterns in which three or more routers are in a loop situation. Triggered updates are an attempt to speed up this convergence.

With triggered updates, when a router changes the metric for a route, it is required to send update messages almost immediately, even if it is not yet time for one of the regular update messages. Some implementations of RIP, specify a small time delay, in order to avoid having triggered updates generate excessive network traffic.

Suppose a router's (C) route to destination network X goes through another router (router G). If an update arrives from router G, the receiving router is required to believe the new information, whether the new metric is higher or lower than the old one. If the result is a change in metric, then the receiving router will send triggered updates to all routers directly connected to it. They in turn may each send updates to their neighbours. This results is a cascade of triggered updates.

However, the only neighbours who will believe the new information are those whose routes for network X go through router C. The other routers will see this as information about a new route that is worse than the one they are already using, and ignore it. The neighbours whose routes go through C will update their metrics and send triggered updates to all of their neighbours. Again, only those neighbours whose routes go through them will pay attention. Thus, the triggered updates will propagate backwards along all paths leading to router B, updating the metrics to infinity.

While the triggered updates are being sent, regular updates may be happening at the same time. Routers that haven't received the triggered update yet will still be sending out information based on the route that no longer exists. It is possible that after the triggered update has gone through a router, it might receive a normal update from one of these routers that hasn't yet been updated. This could cause an incorrect routing entry in that router. If triggered updates happen quickly enough, this is very unlikely.

All implementations of RIP must implement triggered update for deleted routes and may implement triggered updates for new routes or change of routes. RIP implementations must also limit the rate which of triggered updates may be transmitted.

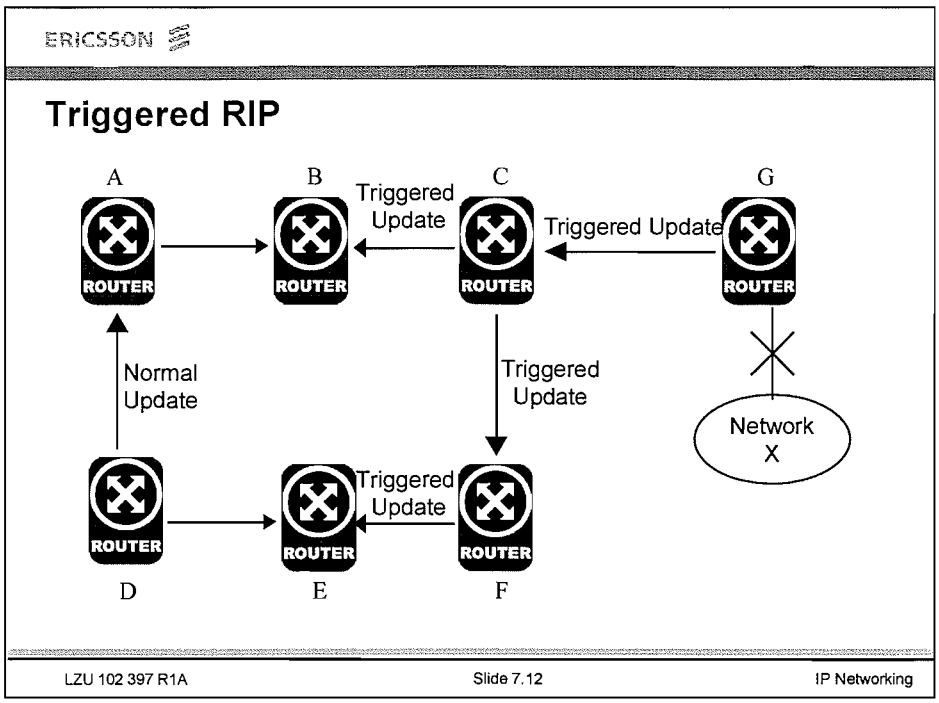


Figure 7-11.

Notes:

TIMERS IN RIP

RIP specifies that all routers must timeout routes they learn via RIP. When a router installs a route in its table, it starts a timer for that route. The timer is restarted whenever the router receives another RIP message advertising that route. The route becomes invalid if a predefined period of time passes without the route being advertised again.

The output process is instructed to generate a complete response to every neighbouring router every 30 seconds. In order to reduce collisions on broadcast networks, implementations are required to take one of two precautions.

- The 30-second updates are triggered by a clock whose rate is not affected by system load or the time required to service the previous update timer.
- The 30-second timer is offset by addition of a small random time each time it is set.

There are two timers associated with each route, a "timeout" and a "garbage collection time". The garbage collection time is often referred to as the hold down timer. Upon expiration of the timeout, the route is no longer valid. However, it is retained in the table for a short time, so that neighbours can be notified that the route has been dropped. Upon expiration of the garbage collection timer, the route is finally removed from the tables.

The timeout is initialized when a route is established, and any time an update message is received for the route. If 180 seconds elapse from the last time the timeout was initialized, the route is considered to have expired and the following takes place.

- The garbage-collection timer is set for 120 seconds.
- The metric for the route is set to 16 (infinity). This causes the route to be removed from service.
- A flag is set noting that this entry has been changed, and the output process is signalled to trigger a response.

Until the garbage-collection timer expires, the route is included in all updates sent by this host, with a metric of 16 (infinity). When the garbage-collection timer expires, the route is deleted from the tables.

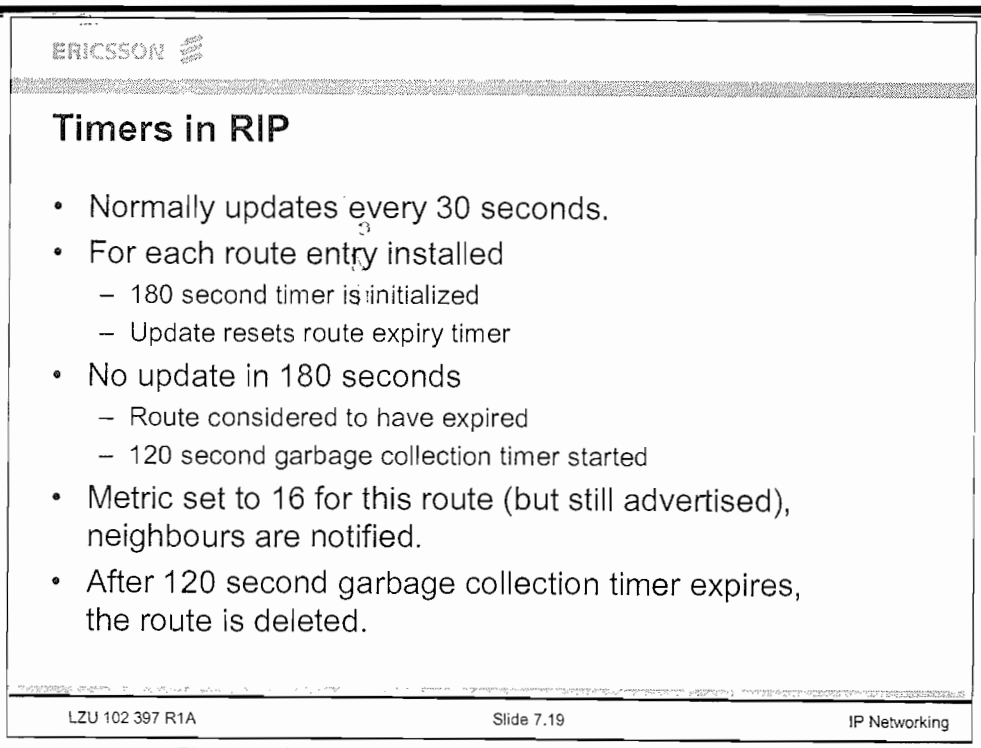


Figure 7-12.

Notes:

ADVANTAGES OF RIP

- RIP automatically creates and maintains a network of routes.
- Since RIP uses a single metric, it is very easy to configure.
- RIP version 2 supports CIDR.
- RIP version 2 has new enhancements added that allow the network administrator to configure routing policies.

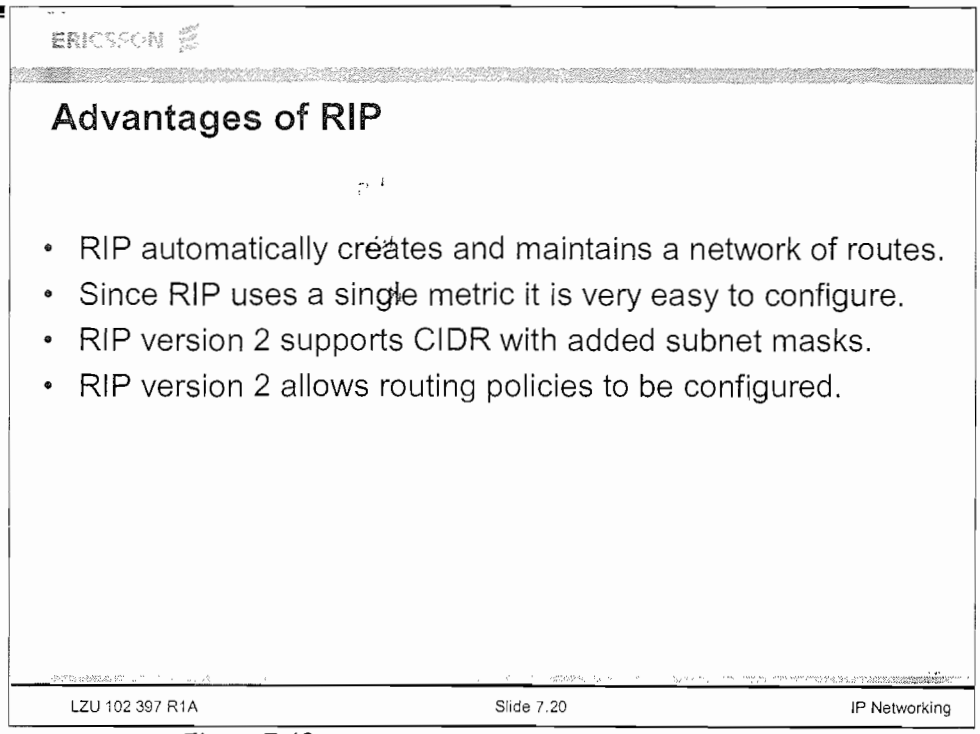


Figure 7-13.

Notes:

DISADVANTAGES OF RIP

- RIP uses hop count as its routing metric, and the value is allowed to range from 1 to 15 only. This limits the diameter of the internetwork to 15 router hops.
- Network administrators cannot take into account such factors as bandwidth or delay when configuring their routing systems. Computing routes on the basis of hop counts has the severe disadvantage that it makes routing relatively static, because routes cannot respond to changes in network load.
- Each routing entry is updated every 30 seconds or so, regardless whether there has been a change in the network topology. Much of this information is redundant. Consequently a lot of bandwidth in taken up with routing updates.
- When a router receives updates advertising networks with equal number of hops it will choose the route that it heard first. This introduces some randomness in the routing process because the timing of updates cannot be predicted. RIP does not support load sharing among routes of equal hops as OSPF does although the protocol does not in theory preclude this. It is simply not done to reduce complexity.

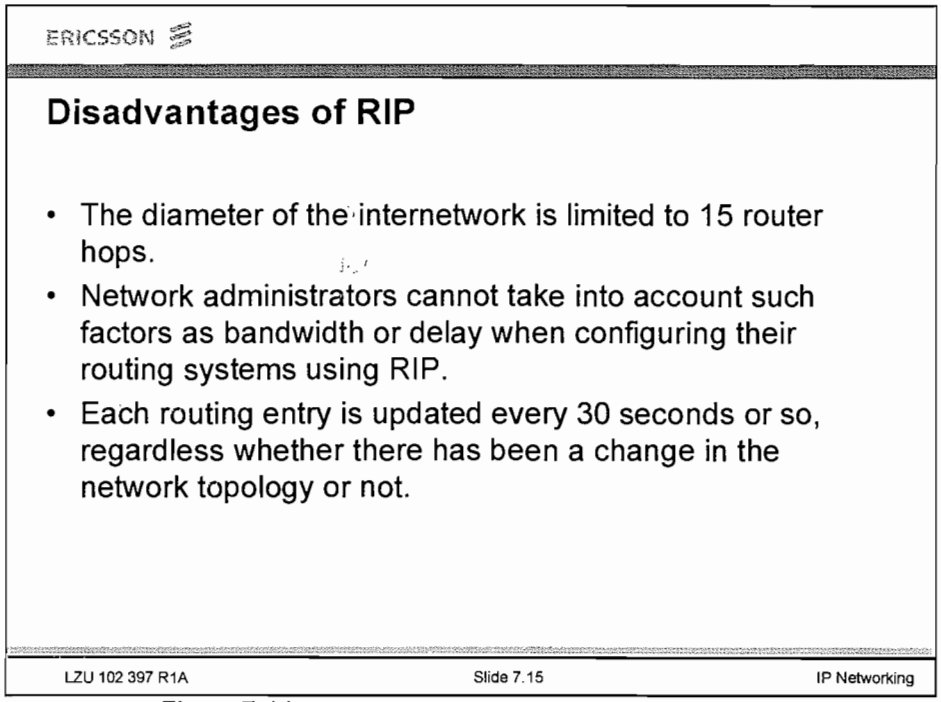


Figure 7-14.

Notes:

Intentionally Blank

8 Open shortest Path First (OSPF)

Intentionally Blank

OPEN SHORTEST PATH FIRST

The development of the Open Shortest Path First (OSPF) routing protocol began in 1987. To understand the goals of the OSPF working group, one needs to consider the nature of the Internet of 1987. It was largely an academic and research network, funded by the US government. Much of the Internet used static routing; Autonomous Systems employing dynamic routing used (Routing Information Protocol) RIP, while External Gateway Protocol (EGP) was used between Autonomous Systems. OSPF version 2 is documented in RFC 2328. Both were experiencing problems. As the size of Autonomous Systems grew and the size of the Internet routing tables increased, the amount of network bandwidth consumed by RIP updates was increasing, and route convergence times were becoming unacceptable as the number of routing changes also increased.

Other initial functional requirements for the OSPF protocol included the following:

- A more descriptive routing metric. A configurable link metric whose value ranges between 1 and 65,535 was chosen. This removed network diameter limitations and allowed factors such as bandwidth, cost and delay to be used when configuring routing systems.
- Equal-cost multipath. OSPF can discover multiple best paths to a given destination. With equal cost multipath, a router potentially has several available next hops towards any given destination.
- Routing hierarchy. This enables us to build very large routing domains, on the order of many thousands of routers.
- Support for more flexible subnetting schemes. OSPF supports variable length subnet masks (VLSMs), whereby a class A, B or C address can be divided into unequal subnets.
- Security. OSFP packets have a space reserved for authentication. By authenticating received OSPF packets, a router would have to be given the correct key before it could join the OSPF routing domain.

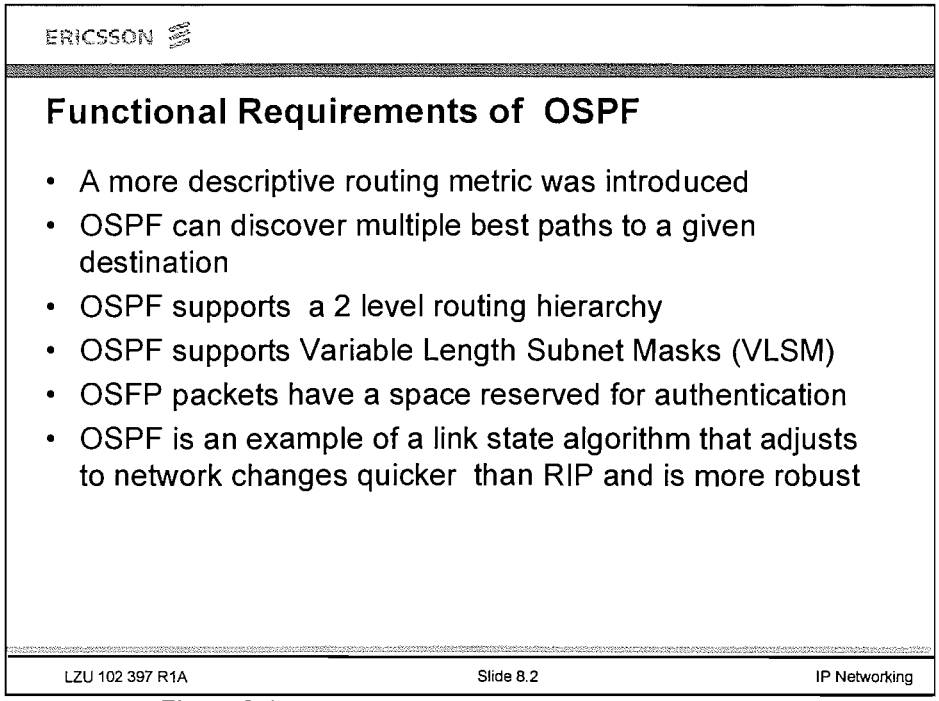


Figure 8-1.

Notes:

LINK STATE PROTOCOL

The Open Shortest Path First (OSPF) routing protocol is a link state algorithm, that adjusts to network changes more quickly than RIP and is more robust. Each router updates the rest of the network with information on the direct connections it has to its neighbours.

OSPF routers advertise their routing information in Link-State Advertisements, or LSAs. OSPF routers broadcast LSAs to their neighbours only when a network change has occurred. They also send an update following a large interval, typically every hour. LSAs are described in detail later in this chapter.

Link-State Advertisements

If a router advertises all of its information in a single LSA, that LSA could get very large. Instead of advertising a single large LSA, OSPF allows the routers to originate multiple smaller LSAs. We call this behaviour LSA fragmentation. OSPF LSAs are smaller in size than the smallest common link Message Transfer Unit (MTUs) found in the Internet. The smallest common MTU is Ethernet's 1,500 bytes. In OSPF, each separate LSA is responsible for advertising each separable piece of routing data. This means that when a small change in the network topology occurs, only the changed routing information is reflooded.

Hierarchical Structure

In order to be able to build large OSPF networks, an OSPF routing domain is split into regions called areas. Details of any particular area are hidden from all other areas. This reduces the size of a router's routing table within an area and enables the size of the overall routing domain to grow larger. In order to advertise information about an area to other areas, the information is reduced. A summary of addresses reachable within an area is passed from area to area.

All areas are connected to a special area called a backbone area. Areas are connected to the backbone using an interface on one of the routers in an area. Forcing all areas to be physically connected to a single backbone area may not be practical. To get around this a logical extension of the backbone area through the configuration of a virtual link is made.

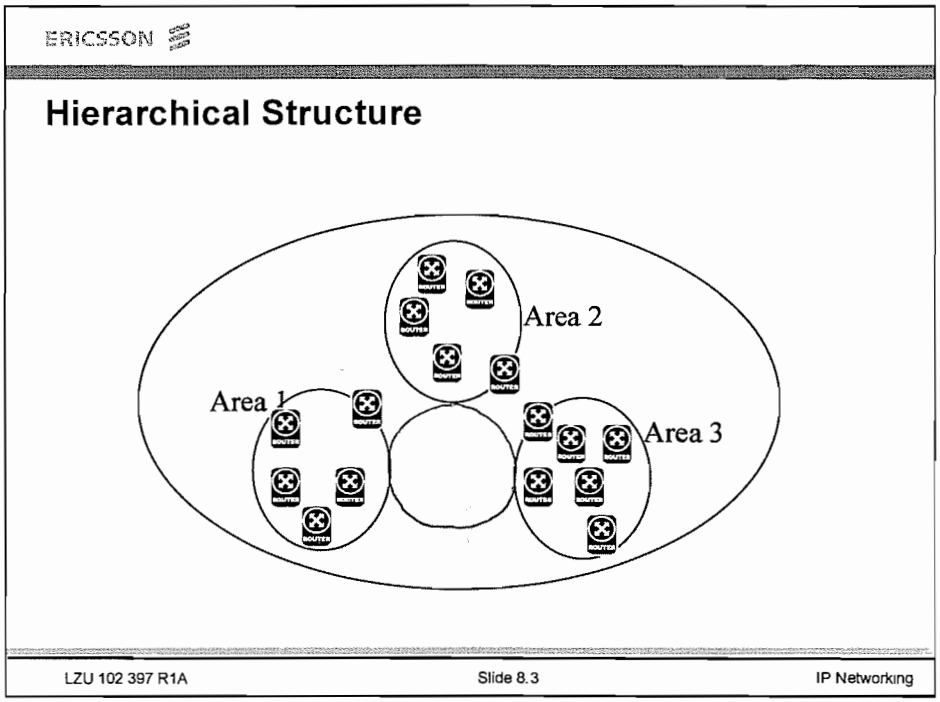


Figure 8-2.

Notes:

HIERARCHICAL ROUTING IN OSPF

OSPF supports a two-level hierarchical routing scheme through the use of OSPF areas. Routers within an area are referred to as level 1 routers. Routers that attach to multiple areas are referred to as level 2 routers. The benefit of hierarchical routing is scalability. Each router does not need to have information on all subnets and routers in the network. It only needs to know the routers and subnets in its own area as well as how to reach an inter area (level 2) router. Level 2 routers are responsible for knowing how to reach all the areas.

An OSPF area is a contiguous collection of router interfaces that depend on OSPF for route calculation, and whose topological detail may be hidden from other portions of the routing domain. Each OSPF area is identified by a 32-bit Area-ID and consists of a collection of network segments interconnected by routers.

When no OSFP areas are configured, each router in the autonomous system has an identical link-state database, leading to an identical graphical representation. A router generates its routing table from this graph by calculating a tree of shortest paths with the router itself as root. The shortest path tree depends on the router doing the calculation. Conventional flat routing is used inside each area. Each area has its own link state database. Each area runs a separate copy of the basic link-state routing algorithm. Detailed knowledge of the area's topology is hidden from other areas.

Conversely, routers internal to a given area know nothing of the detailed topology external to the area. This isolation of knowledge enables the protocol to effect a marked reduction in routing traffic as opposed to treating the entire Autonomous System as a single domain.

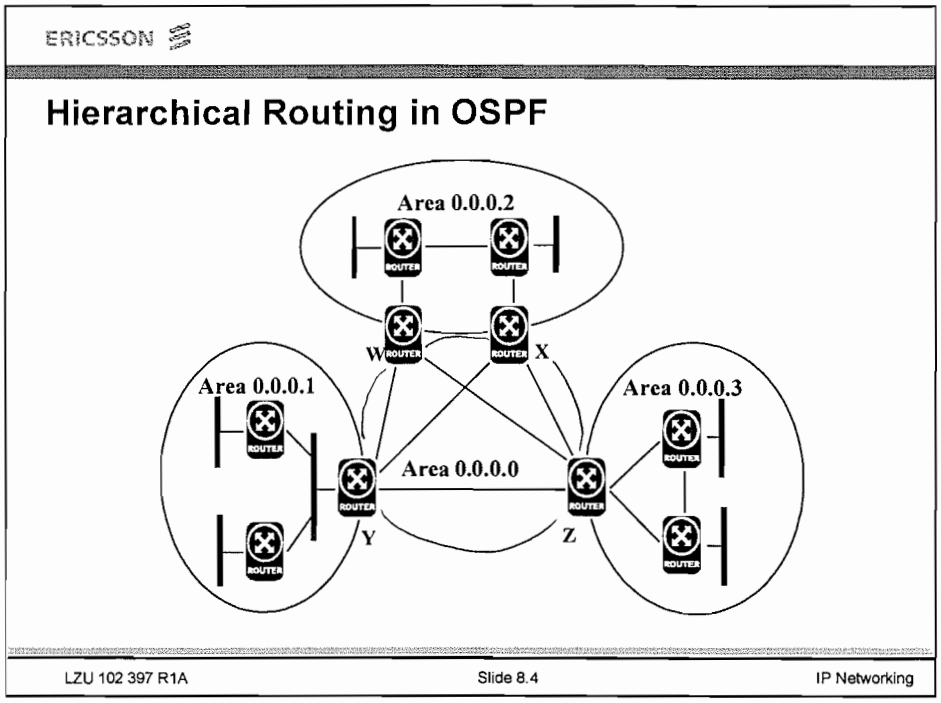


Figure 8-3.

Notes:

Area Border Routers (ABR)

With the introduction of areas, it is no longer true that all routers in the AS have an identical link-state database. A router has a separate link-state database for each area it is connected to. Routers connected to multiple areas are called Area Border Routers (ABR). Two routers belonging to the same area, have, for that area, identical link-state databases. ABRs leak IP addressing information from one area to another in OSPF summary-LSAs. This enables routers in the interior of an area to dynamically discover routes to destinations in other areas. These are called inter-area destinations.

Backbone of the Autonomous System

When an OSPF routing domain is split into areas, all areas are required to attach directly to a special area called the OSPF backbone area. The backbone area always has the Area ID 0.0.0.0. The OSPF backbone always contains all Area Border Routers. The backbone is responsible for distributing routing information between non-backbone areas. ABRs run multiple copies of the basic algorithm, one copy for each attached area. ABRs condense the topological information of their attached areas for distribution to the backbone. The backbone in turn distributes to other areas.

The backbone must be contiguous. However it need not be physically contiguous. Backbone connectivity can be established and maintained through the configuration of virtual links.

Each ABR summarises the topology of its attached non-backbone areas for transmission on the backbone and hence to all other ABRs. An ABR then has the complete topological information concerning the backbone and the area summaries from each of the other ABRs. From this information, the router calculates paths to all inter-area destinations. The router then advertises these paths into the attached areas. This enables the area's routers to pick the best exit router when forwarding traffic to inter-area destinations.

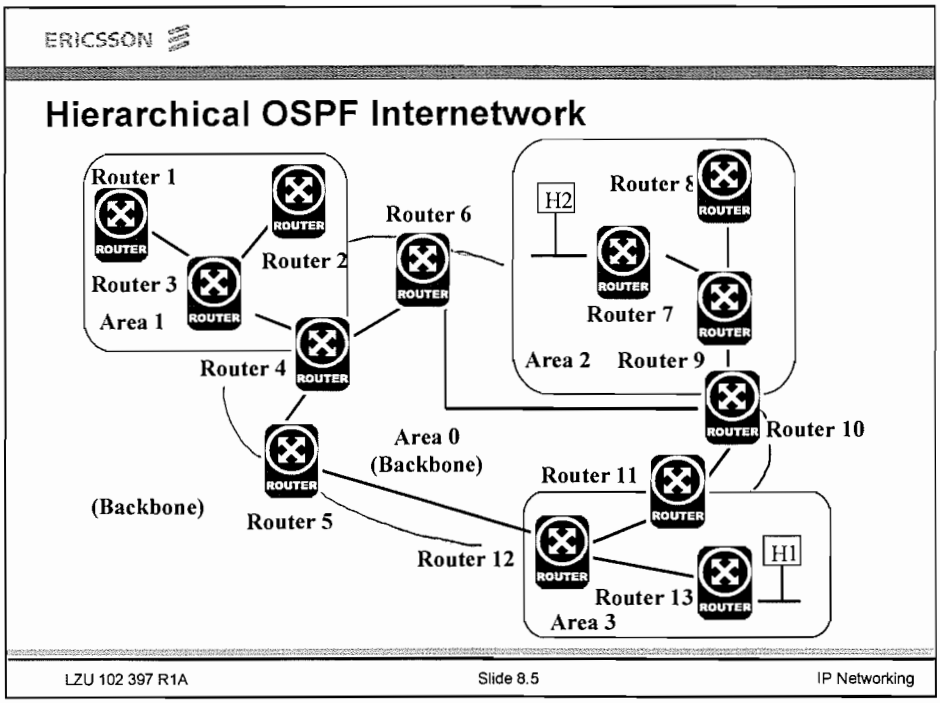


Figure 8-4.

Notes:

Inter-area Routing

When routing a packet between two non-backbone areas the backbone is used. The path that the packet travels is broken into three contiguous pieces; an intra-area path from the source to the ABR, a backbone path between the source and destination areas, and then another intra-area path to the destination. Looking at this in another way, inter-area routing can be pictured as forcing a star configuration on an Autonomous System, with the backbone as hub and each of the non-backbone areas as the spokes.

AS Boundary Routers (ASBR)

When an OSPF routing domain is connected to an external network, a special router known as the Autonomous System Boundary Router (ASBR) is used to interconnect between the OSPF routing domain and the external routing domain. Information is distributed verbatim to every participating router. The paths to each ASBR are known by every router in the AS. ASBRs may be internal or ABRs and may or may not participate in the backbone. The ASBR leaks OSPF summary-LSAs from the external routing domain to the OSPF routing domain and visa versa.

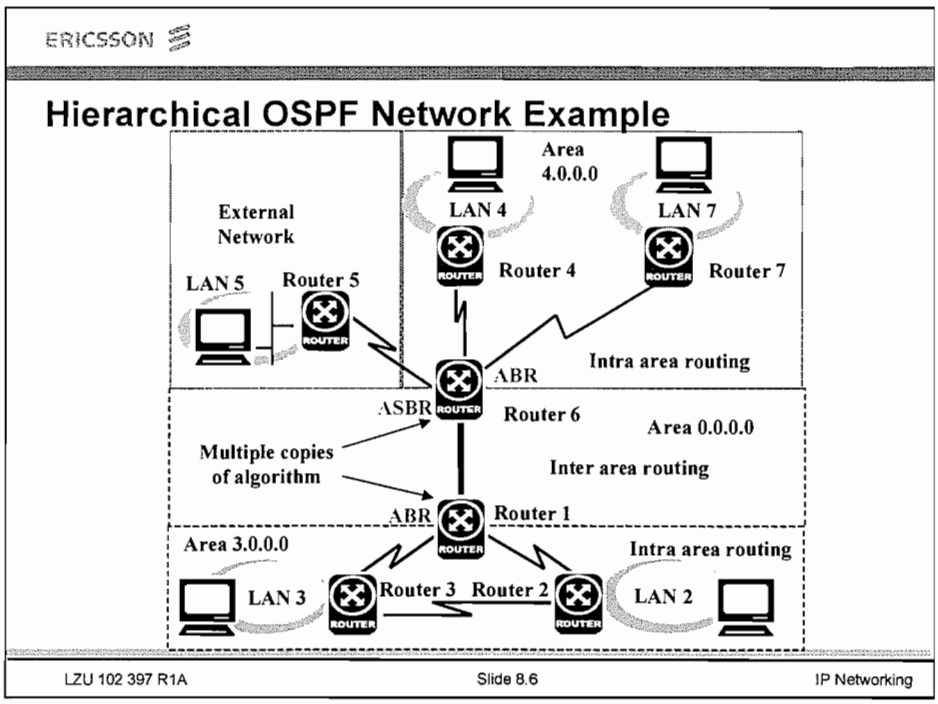


Figure 8-5.

Notes:

OSPF AREA TYPES

OSPF supports three area types. These are:

- Transit Areas
- Stub Areas
- Not So Stubby Areas (NSSA)

Transit Area

A transit area includes any area capable of propagating or originating AS external LSAs. (AS external LSAs are OSPF external-LSAs (Type 5) that are originated by AS Border routers and propagated throughout the OSPF routing domain). The backbone area is always, by definition, a transit area.

Stub Area

In some Autonomous Systems, the majority of the link state database may consist of AS-external-LSAs. An OSPF AS-external-LSA is usually flooded throughout the entire AS. However, OSPF allows certain areas to be configured as stub areas.

Stub areas do not import external route information. OSPF stub areas cannot contain ASBRs. External LSAs (Type 5) are not propagated into the area nor may a stub area originate External LSAs. Instead, network traffic to destinations not local to the area or AS is directed to the closest area border router advertising a default route. There is a default route per area. This reduces the link-state database size and therefore the memory requirements, for a stub area's internal routers.

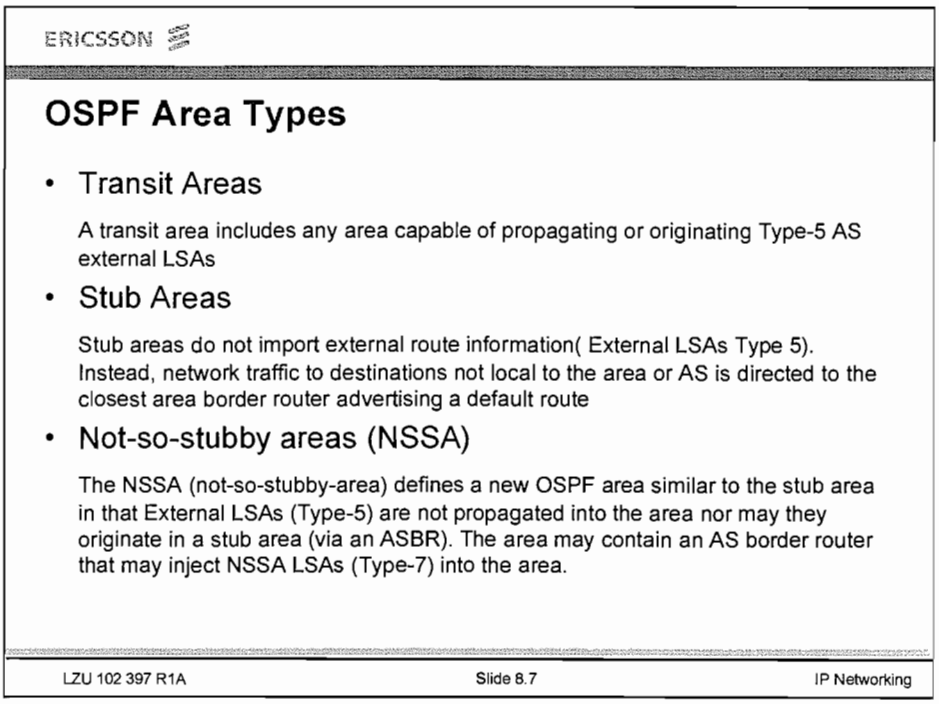


Figure 8-6.

Notes:

Not So Stubby Areas

The NSSA (Not-So-Stubby-Area) defines a new OSPF area similar to the stub area in that external LSAs (Type 5) are not propagated into the area nor may they originate in a stub area (via an ASBR). However, the area may contain an AS border router that may inject NSSA LSAs (Type 7) into the area. These NSSA LSAs (Type 7), which are virtually the same syntax as an external-LSA (Type 5), may then be propagated into other areas by the ABRs as regular external-LSAs (Type 5).

Several configuration options must be completed for an area to be configured as a NSSA. First, the area must be configured as NSSA on all internal routers or routers that have an interface in the area. Any ASBR within the area must be configured to import externally derived routing information (such as RIP, BGP, static routing, and so on). This is done by adding import policies and configuring the router as an ASBR router. If NSSA LSAs (Type 7) are to be propagated into the backbone area, the ABR routers must contain area summarisation policies configured to propagate NSSA LSAs (Type 7) as external LSAs (Type 5) outside of the area. Summarisation policies for networks contained within the area (for network summary-LSAs) are still required for area summarisation.

OSPF Area Summarisation

All areas, regardless of type (transit, stub or NSSA), can perform area summarisation at the ABRs. This means aggregating intra-area routes to a network into a network summary (Type 3) summary LSA. With the introduction of NSSA, network summarisation is now configured for NSSA LSAs (Type 7) that originate within an attached NSSA area.

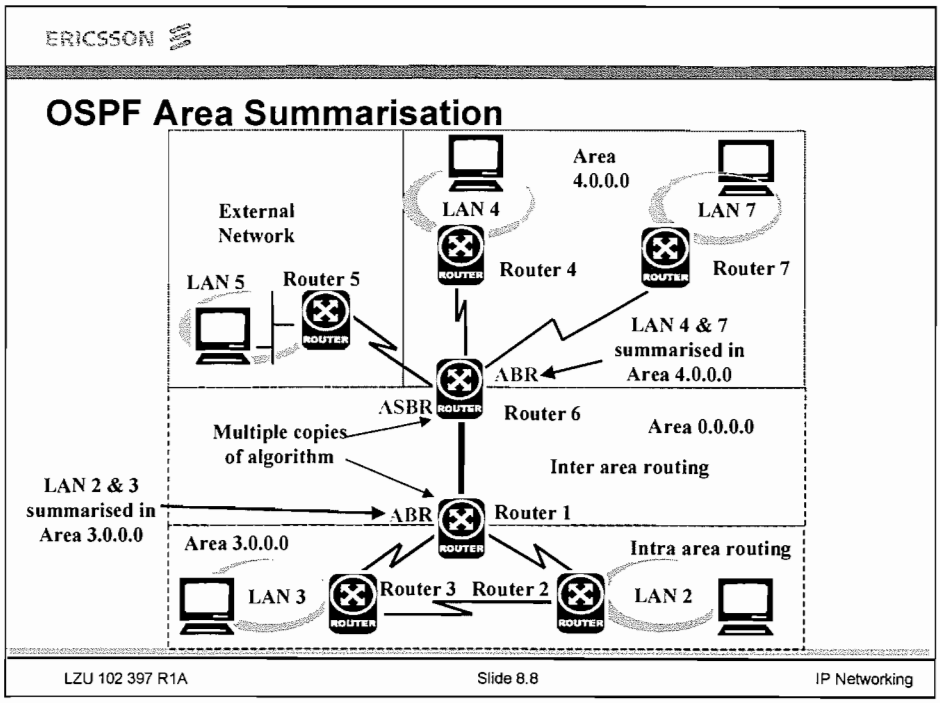


Figure 8-7.

Notes:

OSPF MESSAGE FORMAT

OSPF packets are encapsulated directly into IP packets. The IP protocol number is 89. It was decided not to use TCP as link-state protocols have their own reliability built into their flooding algorithms. Modern routers give preference to routing protocol packets over regular IP data packets, both when being sent and received. OSPF protocol packets should have their IP precedence field set to the value Internetwork Control to accomplish this. There are 5 distinct OSPF packet types. All OSPF packets begin with a standard 24-byte header. This header is described first.

The header consists of the following fields:

- The version field specifies the version of the protocol.
- An OSPF packet type field. There are 5 different types of OSPF protocol messages. These are:

Type	Meaning
1	Hello
2	Database Description
3	Link status Request
4	Link status Update
5	Link state Acknowledgement

- Message length. The length of the OSPF protocol packet in bytes. This length includes the standard OSPF header.
- The OSPF Router ID of the sender, which is normally the IP address of the sending router.
- An OSPF area ID, which enables the receiving router to associate the packet with the proper level of the OSPF hierarchy.
- A packet checksum, which allows the receiving router to check if the packet has been damaged in transit.
- Au Type identifies the authentication procedure to be used for the packet.
- A 64-bit field for use by the authentication scheme.

Routing protocol packets are sent along adjacencies only, with the exception of Hello packets. This means that all routing protocol packets travel a single IP hop, except those sent over virtual links.

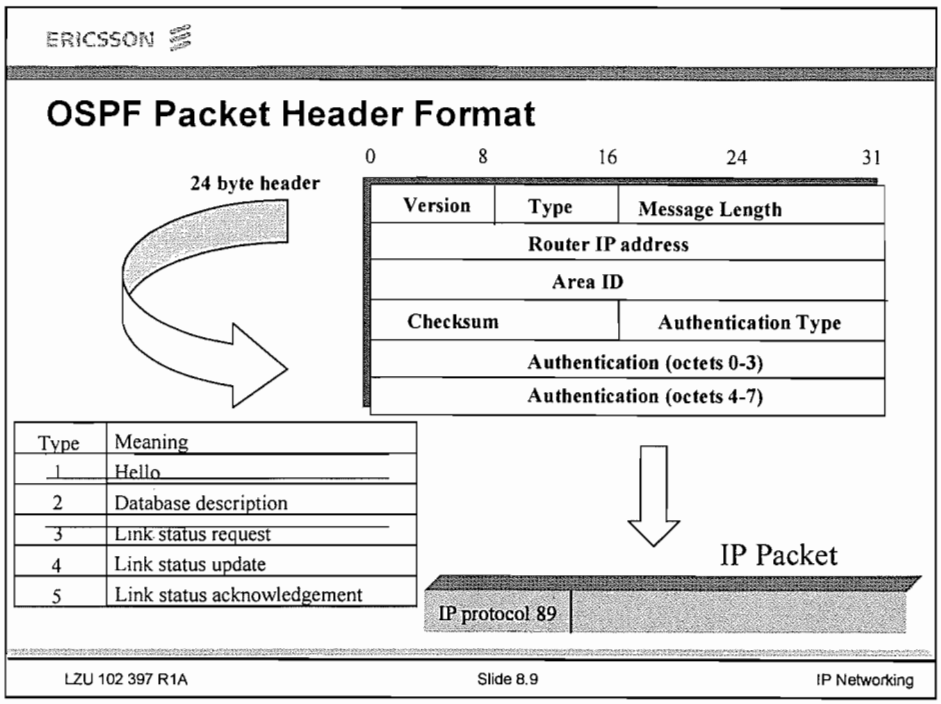


Figure 8-8.

Notes:

Sending OSPF Packets

When a router sends a routing protocol packet it fills the fields of the standard OSPF header as follows:

The version field is set to 2, The Packet Type field is set to Hello, link-state update and so on. The Packet Length is set to indicate the entire length of the OSPF packet, including the standard OSPF header. The Router ID is filled to include the ID of the router sending the packet. The Area ID is updated to indicate the OSPF area. A standard IP 16-bit checksum of the entire OSPF packet, including the 64-bit authentication field is done. Each OSPF packet exchange is authenticated. A different authentication procedure can be used for each IP network or subnet.

The IP destination address is chosen as follows:

On point-to-point networks the IP address is set to AllSPF routers. On all other network types the OSPF packets are sent as unicasts, using the IP address of the adjacent router. The IP source address is the IP address of the sending interface.

Receiving OSPF Packets

When the OSPF packet is received by a router the IP packet must be accepted at the IP level, first of all. The IP checksum must be correct. The IP packets destination address must be the address of the receiving interface. The IP protocol must be OSPF (89).

Next the OSPF packet header is verified. The fields specified in the header must match those configured for the receiving interface. If they do not, the packet is discarded. The version number field must be 2. The Area ID found in the packet must be verified. The packet must be authenticated. This procedure may use one or more Authentication keys, which can be configured on an interface basis. If the authentication procedure fails, the packet is discarded.

The packet is then processed by the relevant protocol, i.e. Hello, database description or flooding protocol.

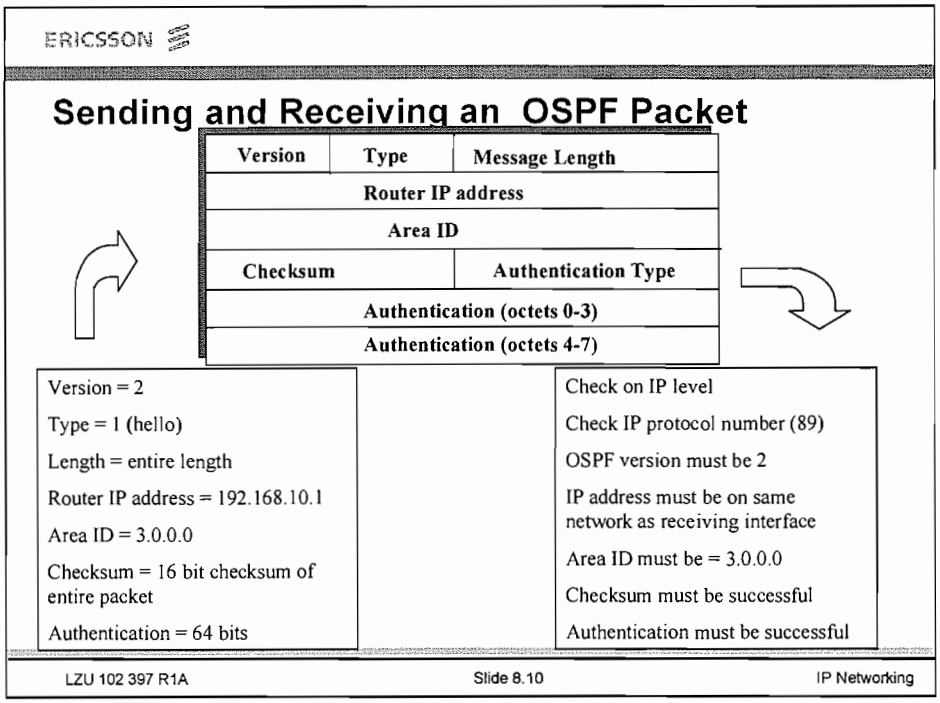


Figure 8-9.

Notes:

THE PROTOCOLS WITHIN OSPF

OSPF is composed of three subprotocols. These are:

- The Hello protocol
- The Exchange protocol
- The Flooding protocol

These protocols are described in the following pages.

The Hello Protocol

OSPF Hello packets have an OSPF header type 1. The Hello protocol is used for two purposes:

- To check that the links are operational
- To elect the Designated Router (DR) and the Backup Designated Router (BDR)

A router discovers its neighbours by periodically sending OSPF Hello packets out all its interfaces. By default, a router sends out these packets at 10-second intervals. The network administrator can configure this interval.

A router learns about its neighbour when it receives a Hello packet from its neighbour. If no Hello packet is received within a certain time interval (40 seconds by default) the router stops advertising the connection to that router and starts routing data packets around the point of failure.

The OSPF Hello protocol also establishes that the neighbouring routers are consistent in the following ways. The Hello protocol ensures that the link is bi-directional.

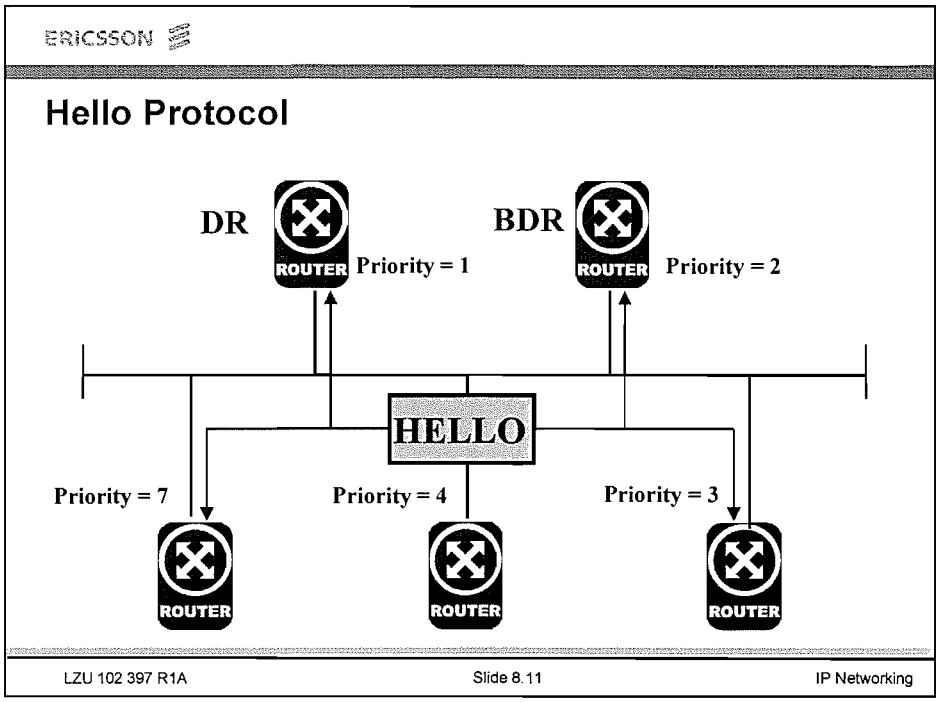


Figure 8-10.

Notes:

DESIGNATED ROUTER

On multi-access networks (networks supporting more than two routers), the Hello protocol elects a Designated Router (DR) and a Backup Designated Router (BDR).

The DR is responsible, among other things, for generating LSAs for the entire multi-access network. DRs allow a reduction in network traffic and in the size of the topological database. The DR and BDR are selected automatically once an OSPF area is configured. A router's Hello packet contains its router priority, which is configurable on a per-interface basis. The DR and BDR are selected using the priority number of the routers in an area. The router with the highest priority is selected as the DR, and the router with the second highest priority is selected as the BDR. If the routers all have identical priorities, then the Router ID is used to select the DR and BDR.

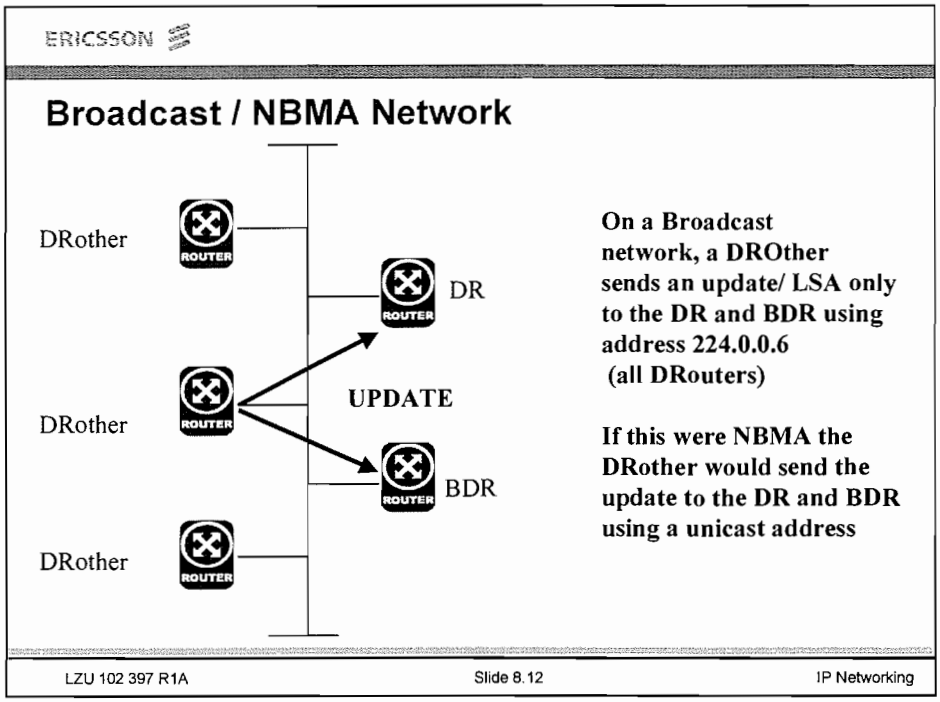


Figure 8-11.

Notes:

Backup Designated Router

The BDR is adjacent to all routers on the network, and becomes the DR when the previous DR fails. If there were no BDR, when a new DR becomes necessary, new adjaciencies would have to be formed between the new DR and all other routers attached to the network.

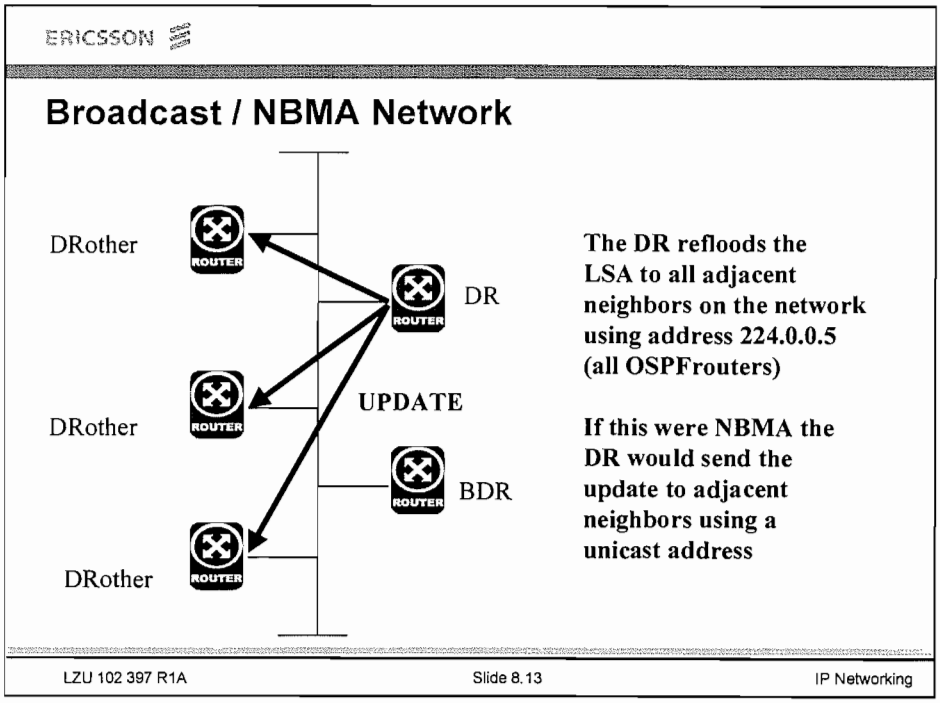


Figure 8-12.

Notes:

Broadcast and NBMA Networks

The Hello protocol works differently on broadcast networks and NBMA networks. On broadcast networks, each router advertises itself by periodically multicasting Hello packets. This allows neighbours to be discovered dynamically.

Routers connected by point-to-point networks, point-to-multipoint networks, and virtual links always become adjacent. On broadcast and Non-Broadcast Multi Access (NBMA) networks, all routers become adjacent to both the DR and the BDR. Topological databases are synchronised between pairs of adjacent routers. Adjacencies control the distribution of routing protocol packets. These packets are sent and received only on adjacencies. Where NBMA networks are in use, the administrator must configure the designated router's IP address into each of its neighbours or give the designated router a list of its neighbours.

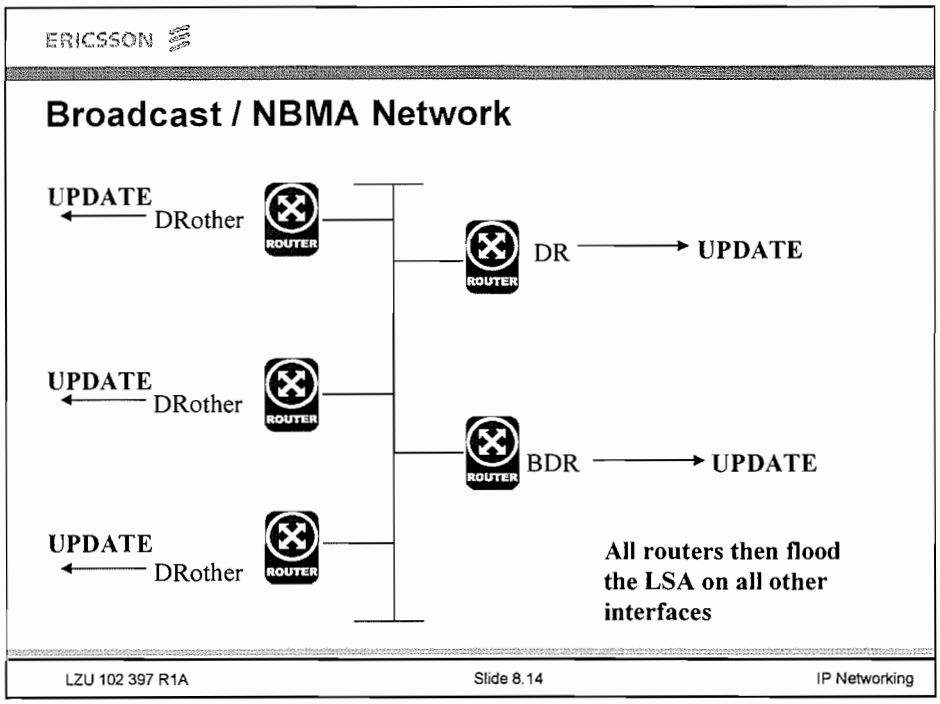


Figure 8-13.

Notes:

DATABASE SYNCHRONISATION

When two routers have established two-way connectivity on a point-to-point link, they must synchronise their databases. The initial synchronisation is performed by the exchange protocol. The flooding protocol is then used to maintain the two databases in synchronisation. The routers must ensure that their link-state databases are synchronised before forwarding traffic over the connection.

The OSPF exchange protocol is asymmetric. The first step of the protocol is to select a "master" and a "slave". After agreeing on these roles, the two routers will exchange the description of their databases, and each lists the records that will be requested at a later stage. The exchange protocol uses database description packets.

Instead of sending the entire database to the neighbour when the connection comes up, an OSPF router sends only its LSA headers, and the neighbours request the most recent LSAs. This is called database exchange. This procedure is more efficient than sending the entire database. The link-state headers are sent in a series of OSPF database description packets. Only one database description packet can be outstanding at any one time; the router sends the next database description packet only when the previous one is acknowledged through reception of a properly sequenced database description packet from the neighbour.

When the entire sequence of database description packets have been received, the router knows the link-state headers of all the LSAs in its neighbours link state database. The router also knows which of its neighbour's LSAs it does not have and which of it's neighbour's LSAs are more recent. The router then sends link state request packets to the neighbour requesting the desired LSAs, and the neighbour responds by flooding the desired LSAs in link state update packets.

Once this process is complete, the routers declare the connection synchronised and advertise it for use by data traffic. At this point the neighbour is said to be fully adjacent to the router.

The router may have Max Age LSAs in its link-state database as database exchange begins. Since Max Age LSAs are in the process of being deleted from the database, the router does not send them in Database Description packets to the neighbour.

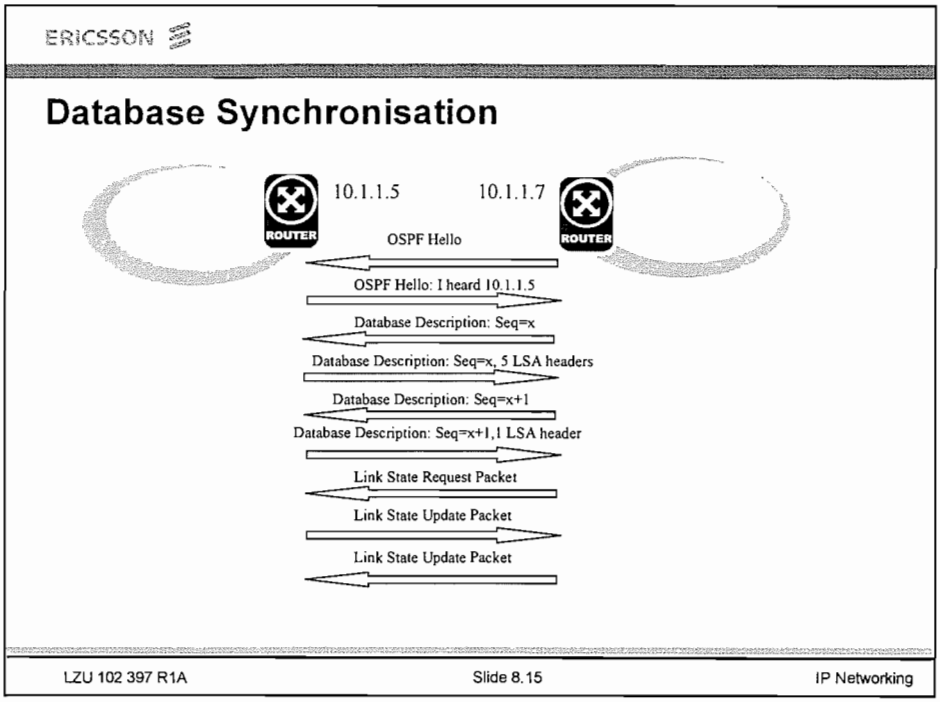


Figure 8-14.

Notes:

The Hello Protocol Message

Hello packets are OSPF packets with the type field set to 1. The Hello packet format is illustrated in the following diagram. The fields in the packed are described below:

- Network Mask: this is the network mask associated with the interface.
- Options: the optional capabilities supported by the router.
- Hello Interval: the number of seconds between this router's Hello packets.
- Rtr Pri: this is the router's priority. This is used in the election of the DR and BDR. If set to 0, the router will be ineligible to become (Backup) Designated Router.
- Router Dead Interval: the number of seconds before declaring a silent router down.
- Designated Router: this is the identity of the DR in this network. The Designated Router is identified by its IP interface address on the network. This field is set to 0.0.0.0 when there is no DR.
- Backup Designated Router: this is the identity of the BDR in the network. The Backup Designated Router is identified by its IP interface address on the network. This field is set to 0.0.0.0 when there is no BDR.
- Neighbour: the Router IDs of each router from whom valid Hello packets have been seen recently on the network. Recently means in the last Router Dead Interval seconds.

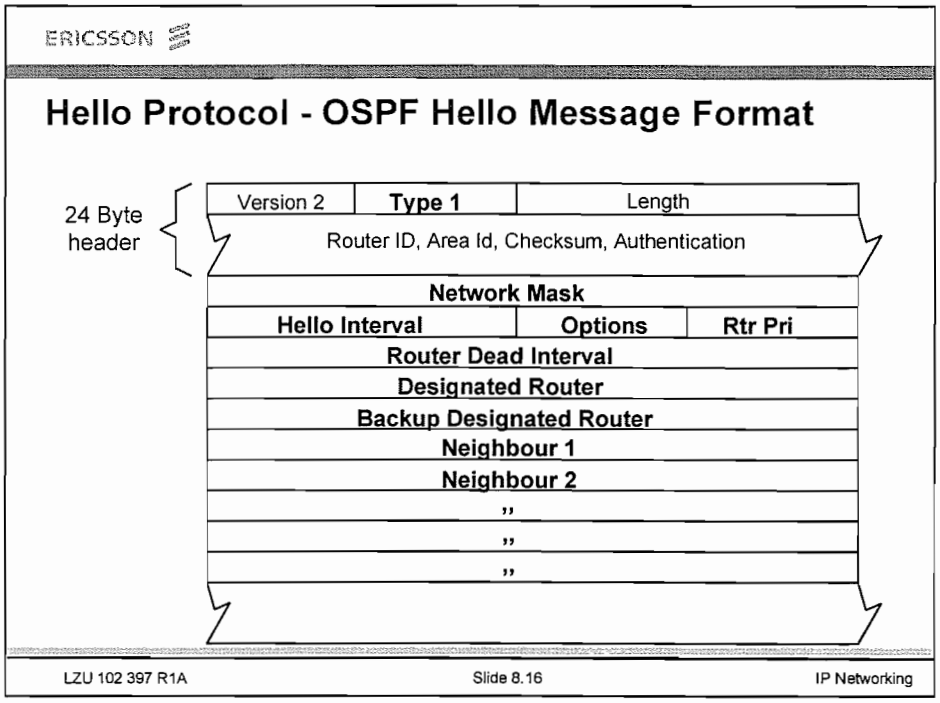


Figure 8-15.

Notes:

OSPF Database Description (Type 2)

The slave acknowledges each database description message with a response. Because it can be large, the topology database may be divided into several messages using the I and M bits. Bit I is set to 1 in the initial message; bit M is set to 1 if additional messages follow. Bit S indicates whether a message was sent by a master (1) or by a slave (0).

- Interface MTU: the size in bytes of the largest IP datagram that can be sent out the associated interface, without fragmentation.
- Options: the optional capabilities supported by the router.
- I-bit: the Init bit. When set to 1, this packet is the first in the sequence of Database Description Packets.
- M-bit: the More bit. When set to 1, it indicates that more database Description Packets are to follow.
- S-bit: the Master/Slave bit. When set to 1, it indicates that the router is the master during the Database Exchange process. Otherwise, the router is the slave.
- DD sequence number is used to sequence the collection of Database Description Packets. The initial value (indicated by the Init bit being set) should be unique. The DD sequence number then increments until the complete database description has been sent.

The fields from Link Age to Length describe one link in the network topology. They are repeated for each link.

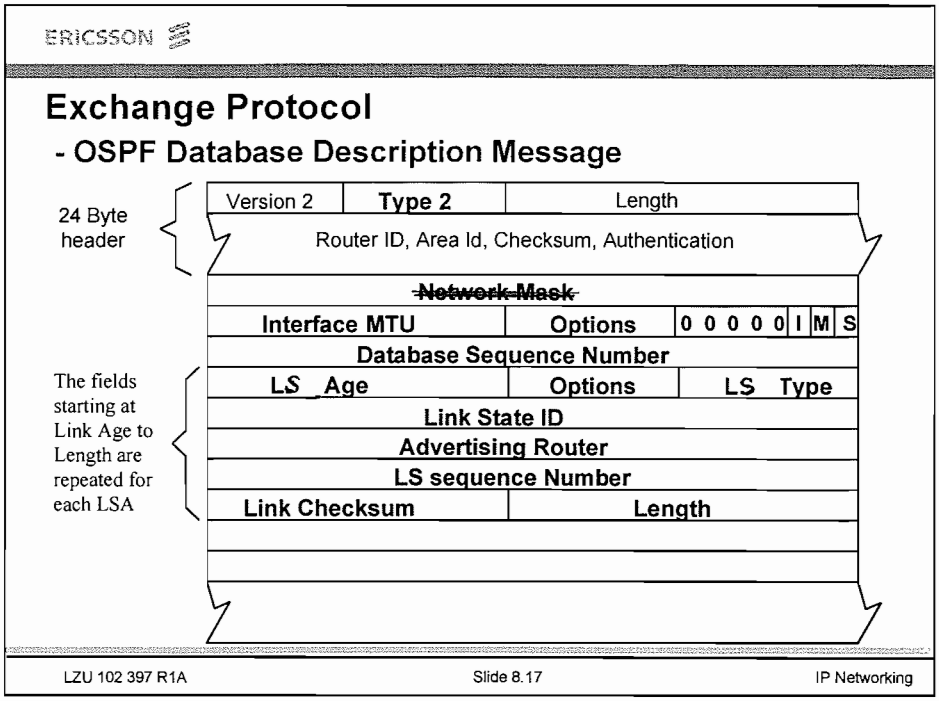


Figure 8-16.

Notes:

OSPF Link Status Request Message (Type 3)

After exchanging database description messages with a neighbour, a router may discover that parts of its database are out of date. To request that the neighbour supplies updated information, the router sends a link status request message. The message lists specific links. The neighbour responds with the most up-to-date information it has about those links. The three fields, Link Type, Link Id and Advertising Router are repeated for each link. More than one request message is sent when the list of requests is long.

Link Status request packets are OSPF packets with the type field set to 3. A router that sends a Link State request packet knows exactly the precise instance of the database it is requesting. The router may receive even more recent instances in response.

Each LSA requested is specified by its LS type, Links State ID, and Advertising Router.

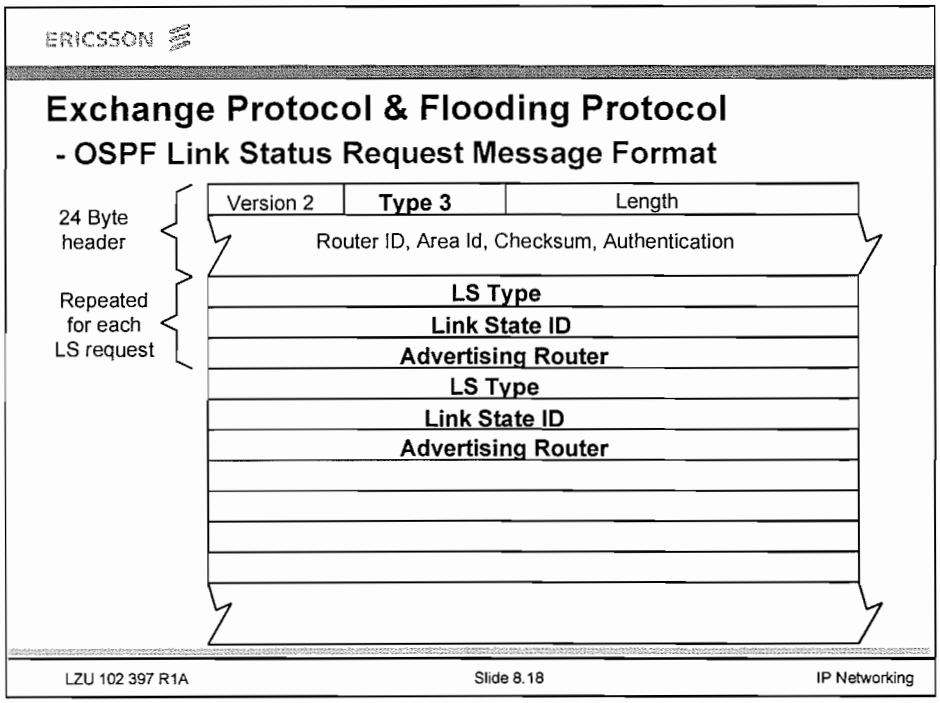


Figure 8-17.

Notes:

OSPF Link Status Update Message (Type 4)

Each link status update message has a header format as shown in the figure. Link State Update packets are OSPF packets with the type field set to 4. These OSPF packets implement the flooding protocol.

The flooding protocol is used to maintain the two databases in adjacent routers in synchronisation.

When a link changes state, the router responsible for that link will issue a new version of the link state. This is carried in link state update message.

The OSPF header is followed by an indication of the number of advertisements and by the link state advertisements themselves.

The values used in the header are the same as in the database description message. The body of the Link State Update packet consists of a list of LSAs.

Each LSA begins with a common 20 byte header. This is illustrated later.

LSA packets are explicitly acknowledged. This is achieved through the sending and receiving of Link State Acknowledgement packets.

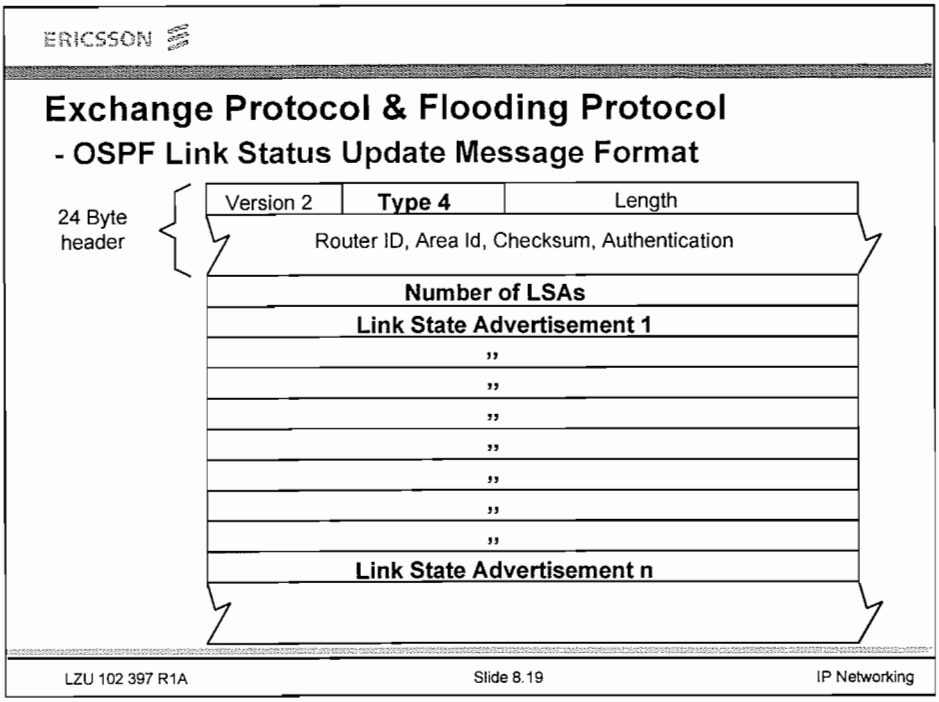


Figure 8-18.

Notes:

Link State Acknowledgement (Type 5)

Link State Acknowledgment Packets are OSPF packet type 5. To make the flooding of LSAs reliable, flooded LSAs are explicitly acknowledged. This acknowledgment is accomplished through the sending and receiving of Link State Acknowledgment packets. Multiple LSAs can be acknowledged in a single Link State Acknowledgment packet.

Depending on the state of the sending interface and the sender of the corresponding Link State Update packet, a Link State Acknowledgment packet is sent either to the multicast address AllSPFRouters, to the multicast address AllDRouters, or as a unicast.

The format of this packet is similar to that of the Data Description packet. The body of both packets is simply a list of LSA headers.

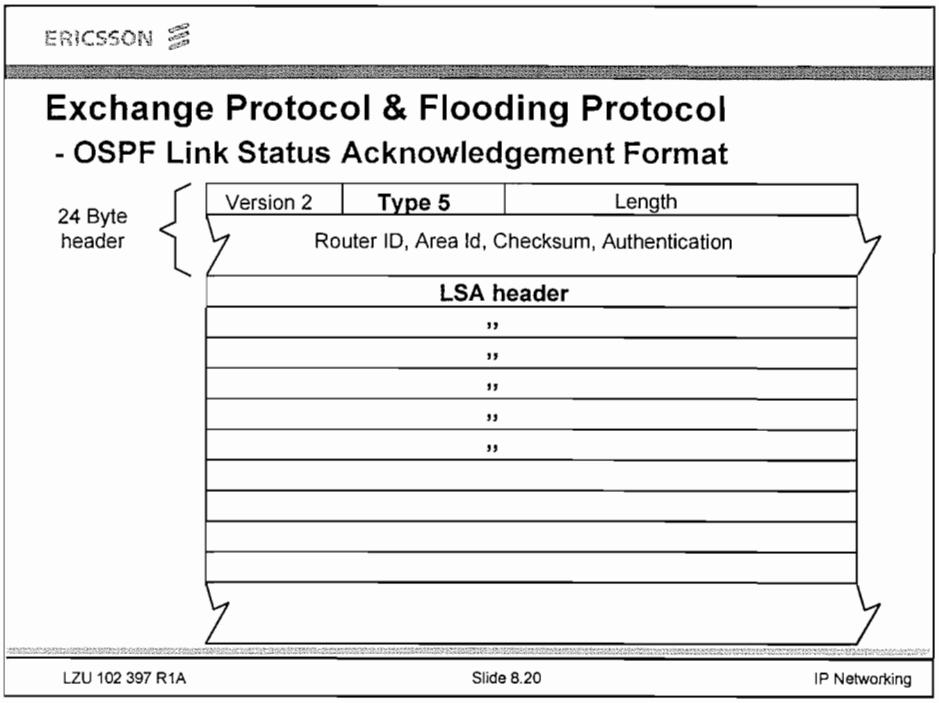


Figure 8-19.

Notes:

LINK STATE ADVERTISEMENTS

The diagram opposite summarises the different types of LSAs in OSPF.

LSA header

All LSAs begin with a common 20 byte header. The LSA contains enough information to uniquely identify the LSA (LS type, Link State ID, and Advertising Router). Multiple instances of the LSA may exist in the routing domain at the same time. It is then necessary to determine which instance is more recent. This is accomplished by examining the LS Age, LS sequence number and LS checksum fields that are also contained in the LS header.

LS age: this is the time in seconds since the LSA was originated.

LS type: this specifies the LSA type. Each LSA type has a separate advertisement format.

Link State ID: this identifies the portion of the Internet environment that is being described by the LSA. This depends on the LSA type.

Advertising Router: this is the router ID of the router that originated the LSA.

LS Sequence number: successive instances of an LSA are given successive LS sequence numbers.

LS checksum: this is a checksum performed on the complete contents of the LSA including the LSA header but excluding the LS age field.

Length: this is the length in bytes of the LSA and includes the 20 byte header.

ERICSSON

Link state types

LS type	LS name	Description
1	Router-LSA	Originated by all routers. This LSA describes the collected states of the router's interfaces to an area. This LSA is flooded throughout a single area only.
2	Network-LSA	Originated for broadcast and NMBA networks by the DR. This LSA contains the list of routers connected to the network. This LSA is flooded throughout a single area only.
3 / 4	Summary -LSA	Originated by ABRs and flooded throughout the LSAs associated area. Each summary-LSA describes a route to a destination outside the area, yet still inside the AS. Type 3 LSAs describe routes to networks. Type 4 LSAs describe routes to AS boundary routers.
5	AS- external -LSA	Originated by ASBR routers, and flooded throughout the AS. Each AS-external-LSA describes a route to a destination in another AS. Default routes can also be described by AS-external-LSAs.
7	NSSA External LSAs	External routes imported by NSSA AS boundary routers. An NSSA router indicates it is an NSSA ASBR by setting the E-bit in its router-LSA.

LZU 102 397 R1A Slide 8.21 IP Networking

Figure 8-20.

Notes:

The Link State Database

The collection of all OSPF LSAs is called a link-state database. Each OSPF router has an identical link-state database. The link state database contains a map of the entire network and can be a powerful debugging tool. By examining the link state database, a system administrator can observe the state of all the routes in the network. For example, one can see which parts of the network are changing the most, by examining the sequence numbers and the LS Age field, in the databases. Links that are frequently changing will have constantly changing sequence numbers and the LS Age field will never be large. The link state record header and the various types of record contents are described in the following paragraphs.

There are several types of link state records, but all records share the same link state advertisement header. The advertising router is identified by one of its IP addresses, which is selected as OSPF identifier for that router. The fields of the header are described below:

- The Link Type describes a link.
- The Link ID field gives identification for the link (which can be the IP address of a router or a network, depending on the link type).
- The Advertising Router field specifies the address of the router advertising this link.
- The Link Sequence Number contains an integer generated by the advertising router to ensure that messages are not missed or received out of order.
- The Link Checksum provides further assurance that the link information has not been corrupted.
- The Link Age gives the time in seconds since the link was established.

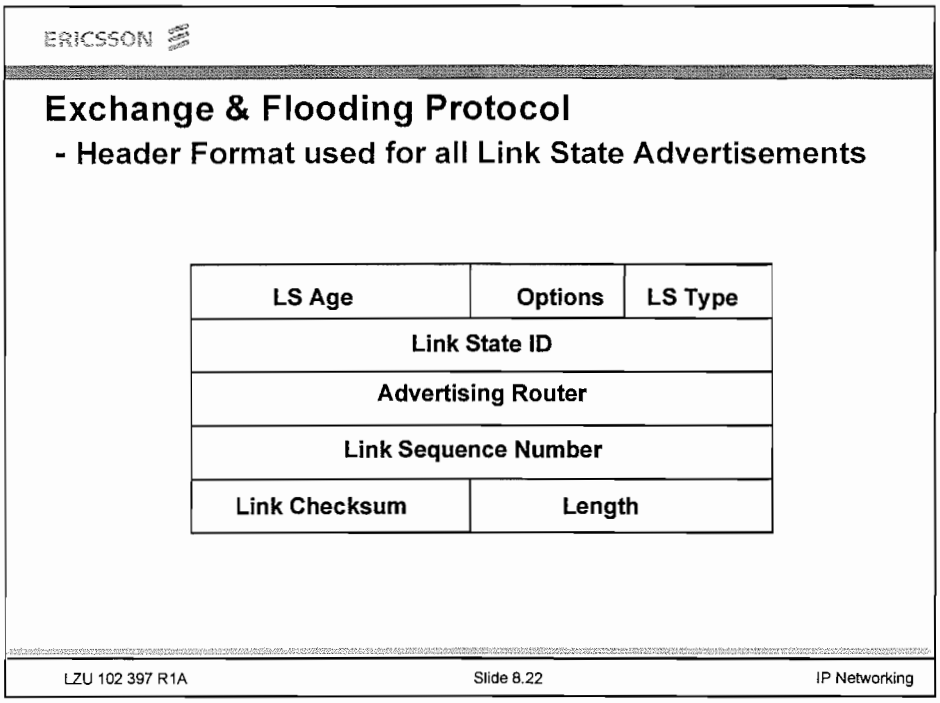


Figure 8-21.

Notes:

The Router Link State Record

The router links state record (link state type 1) summarises all the links that start from the advertising router. For each link, there is a link ID, link data, and a link type.

The content starts with a 32-bit word specifying the number of links and the type of router, followed by a set of link descriptions.

The link ID specifies the OSPF's router ID.

The link type provides a description of the router link:

- Type 1 is a point to point connection to another router.
- Type 2 is a connection to a transit network.
- Type 3 is a connection to a stub network.
- Type 4 is a virtual link.

All advertisements must include the metric of the link for the default Type of Service (TOS). They may also include other metrics for other TOS.

The V bit is for virtual link endpoint

When bit E is set, the router is an AS Boundary Router (ASBR).

When bit B is set, the router is an Area Border Router (ABR).

ERICSSON

The OSPF Database
- Router Links State Record

- - 0 - - VEB	- - - 0 - - - -	Number of links
Link ID		
Link Data		
Type	# TOS=0	Metric

LZU 102 397 R1A Slide 8.32 IP Networking

Figure 8-22.

Notes:

The Network Links State Record

The network links state record (link state type 2) are advertised by Designated Routers for transit networks.

The LS record describes all routers attached to the network, including the DR itself.

The distance from the network to all attached routers is zero. This is why metric fields need not be specified in the network-LSAs.

The content of the record is the 32-bit network or subnet mask for the network followed by the OSPF identifier of all the attached routers.

Only the routers that are fully adjacent to the DR are listed. The number of routers included can be deduced from the LSA header's length field.

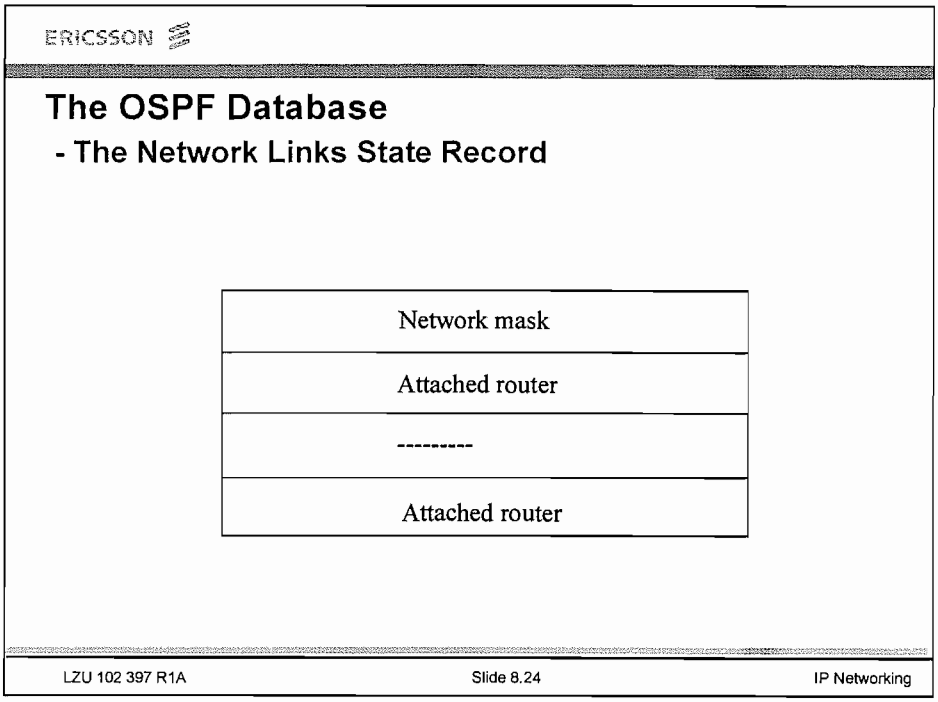


Figure 8-23.

Notes:

The Summary Link State Record

Summary links for IP networks (link state type = 3) and for border routers (link state type = 4) are advertised by Area Border Routers (ABR).

Although these routers may advertise several summary links, they will not pack them in a single advertisement, but will issue one separate advertisement for each destination.

The Link State ID is the IP network or subnet number.

The mask is that of the network or subnet.

ERICSSON

The OSPF Database

- The Summary Links State Record

Network Mask	
0	Metric
TOS = 0	TOS Metric = 0

LZU 102 397 R1A Slide 8.34 IP Networking

Figure 8-24.

Notes:

The External Links State Record

External links are advertised by ASBRs and describe links external to the AS.

The link state ID (in the standard LSA header) is the IP network or subnet number of the destination. The content is a 32-bit mask, followed by a set of metrics. External routes are acquired by the border routers through external gateway protocols such as EGP and BGP. These protocols do not necessary provide estimations of distances in units comparable to the metrics of OSPF.

AS external-LSAs are also used to advertise a default route. Default routes are used when no specific route exists to the destination. When describing a default route, the LINK State ID is always set to 0.0.0.0 and the network mask is set to 0.0.0.0.

The network mask is the IP address mask for the advertised destination. For example, when advertising a class A network the mask is 255.0.0.0. The TOS field includes an E bit at position 0 and a 32-bit "external flag" follows the metric. When the E bit is set to 0, this means that the metric is comparable to those used by OSPF, and the metric can be added to compute the cost of the path to the destination through the ABR. When the E-bit is set it indicates that the metric is not comparable to the internal metrics and should be considered larger than any internal route.

The Forwarding Address is the address to which data traffic will be forwarded.

ERICSSON

The OSPF Database

- The External Links State Record

Network Mask		
E	0	Metric
Forwarding address		
External route tag		
E	TOS = 0	TOS Metric = 0
Forwarding address		
External route tag		

LZU 102 397 R1A Slide 8.35 IP Networking

Figure 8-25.

Notes:

CALCULATION OF THE ROUTING TABLE

This section details the OSPF routing table calculation. Using its attached area's link state databases as input, a router runs the following algorithm, building its routing table step by step.

At each step, the router must access individual pieces of the link state databases (for example, a router-LSA originated by a certain router). This access is performed by the lookup function.

The lookup function may return an LSA whose LS age is equal to MaxAge.

Such an LSA should not be used in the routing table calculation.

The entire process can be summarised in the following steps:

The present routing table is invalidated. The routing table is built again from scratch. The old routing table is saved so that changes in routing table entries can be identified.

The intra-area routes are calculated by building the shortest path tree for each attached area.

The inter-area routes are calculated, through the examination of summary- LSAs. If the router is attached to multiple areas (that is, it is an ABR), only backbone summary- LSAs are examined.

In ABRs connecting to one or more transit areas (that is, non-backbone areas whose transit capability is found to be true), the transit area's summary-LSAs are examined to see whether better paths exist using the transit areas than were found in steps 2 and 3 above.

Routes to external destinations are calculated, through examination of AS external- LSAs.

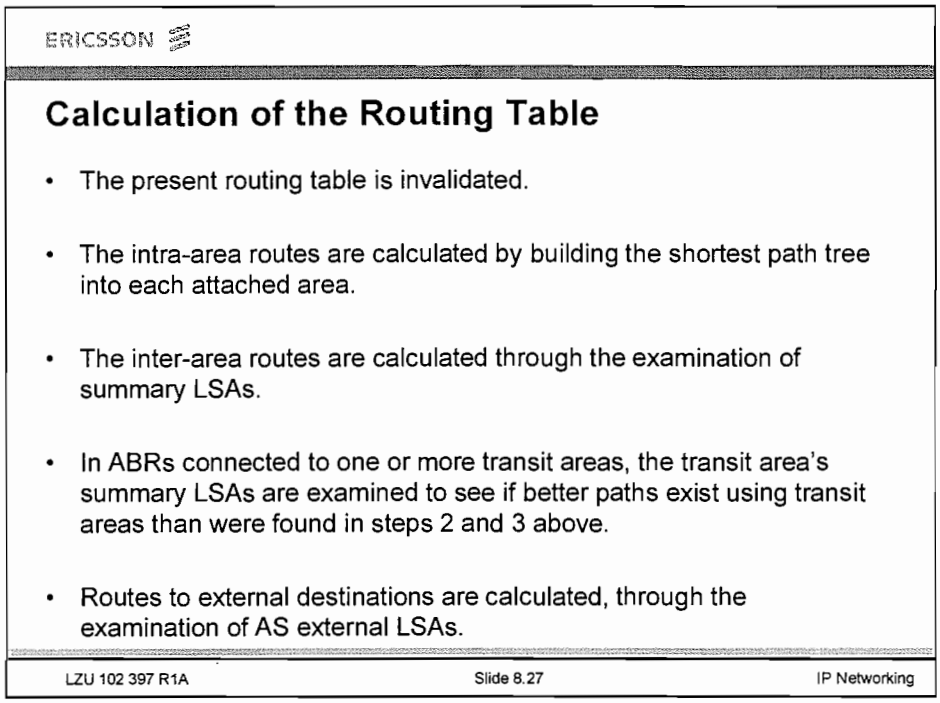


Figure 8-26.

Notes:

Advantages of OSPF

- OSPF is a standard protocol that all vendors can implement interoperabaly.
- OSPF provides a rapid, deterministic calculation of internet routes.
- It facilitates separate administration of different parts of the internet.
- It facilitates hiding of detailed information about the internet.
- OSPF uses detailed information about the internet to optimise route calculations through the use of metrics. Unlike RIP, which is based entirely on hop count, OSPF uses a cost metric to select the best route. Network administrators can define a preferred path by assigning interface costs when configuring OSPF.
- With OSPF, one can isolate misconfigured or malfunctioning routers in the internet and route around them.
- OSPF provides for the effective use of information derived from other routing protocols, even those that use metrics that OSPF does not understand.

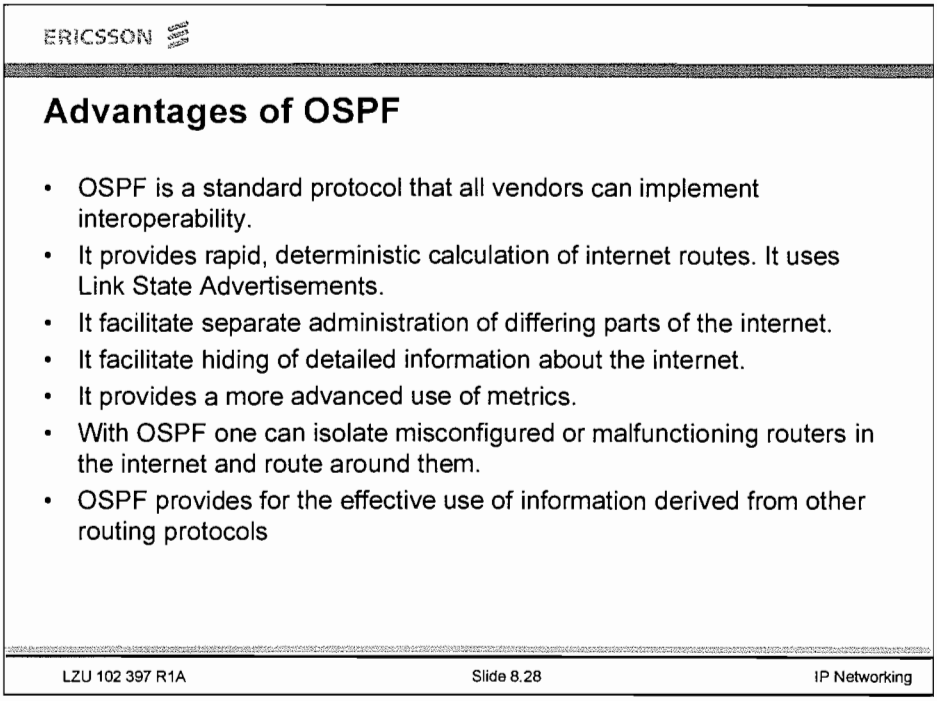


Figure 8-27.

Notes:

Disadvantages of OSPF

- Link-state protocols use large amounts of router memory to store topological databases, as each router keeps a map of the entire network.
- When a network experiences frequent changes, link-state routers use a large proportion of network bandwidth sending out LSAs for each network change.
- After receiving a new LSA, the router must run the Shortest Path First (SPF) algorithm and generate a new routing table. The process places heavy demands on the router's CPU.

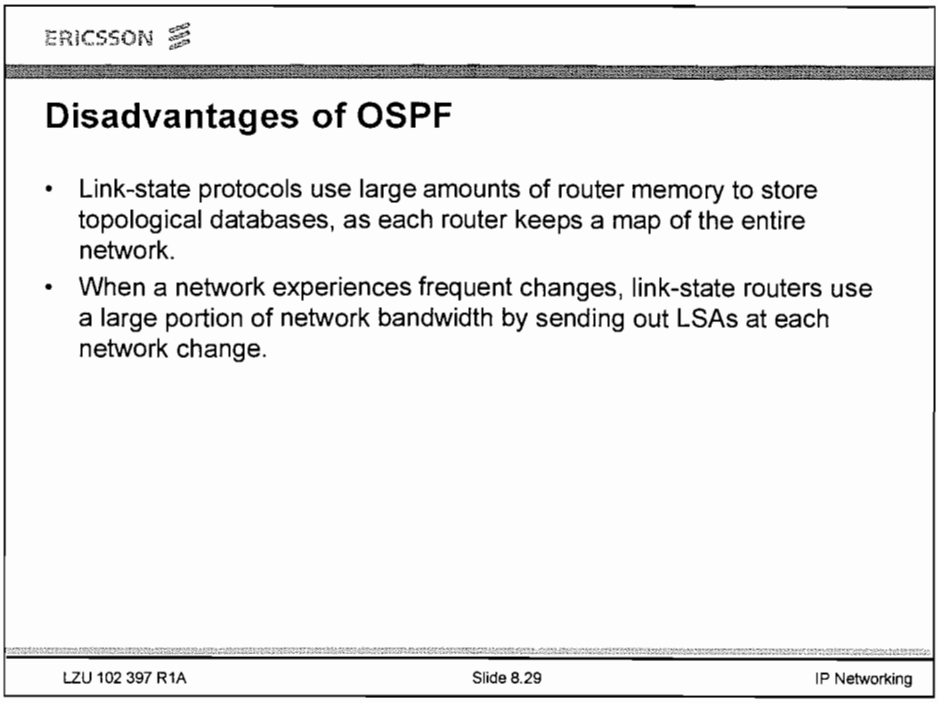


Figure 8-28.

Notes:

Intentionally Blank