Training Course for CWR Internals

Networking

Network Protocols for Windows NT

Protocol Families

Network Protocol Basics

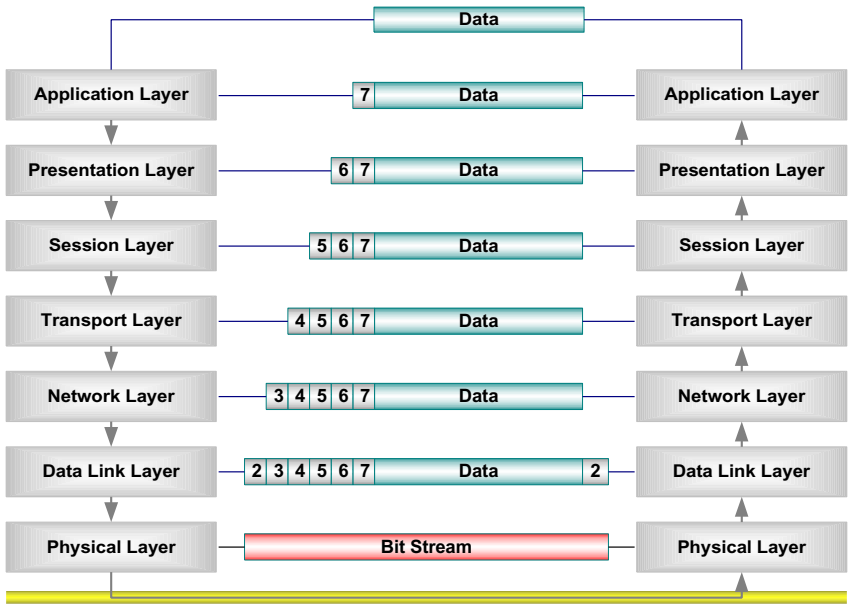
- LAN network traffic is devided into **Packets**
- Several communication layers are needed for a controlled and error-free end-to-end communication between software processes on stations within a network
- Identical layers in different network stations communicate in a defined communication language via a so-called "**Protocol**".
- Each communication layer adds is own protocol control information as header (or trailer) to the payload.
- The control information of a layer will be interpreted and then removed by the corresponding layer on the receiving side.
- The combination of layers which implement a complete communication between applications on different machines are called a „**Protocol Stack**"

Protocol Stacks

- Different protocol stacks implement similar functions and services. However, they use different „languages"
- There are communication incompatibilities between the different protocols.
- Different protocols can co-exist on the same physical network without interferring with each other
- Protocols are distinguished by internationally agreed 16 bit type codes (e.g.: IP = 0x800, ARP = 0x806, XTP = 0x817D, ...)
- The lower 4 layers are the soo called „Transport System"
- The Transport System provides procedures (protocols) for a networkwide process-to-process (software applications) communication. Examples are: TCP/IP, UDP/IP, XTP, ISO-TP4, SPX/IPX, NetBEUI, AppleTalk ATP/DDP

The OSI 7 Layer Model

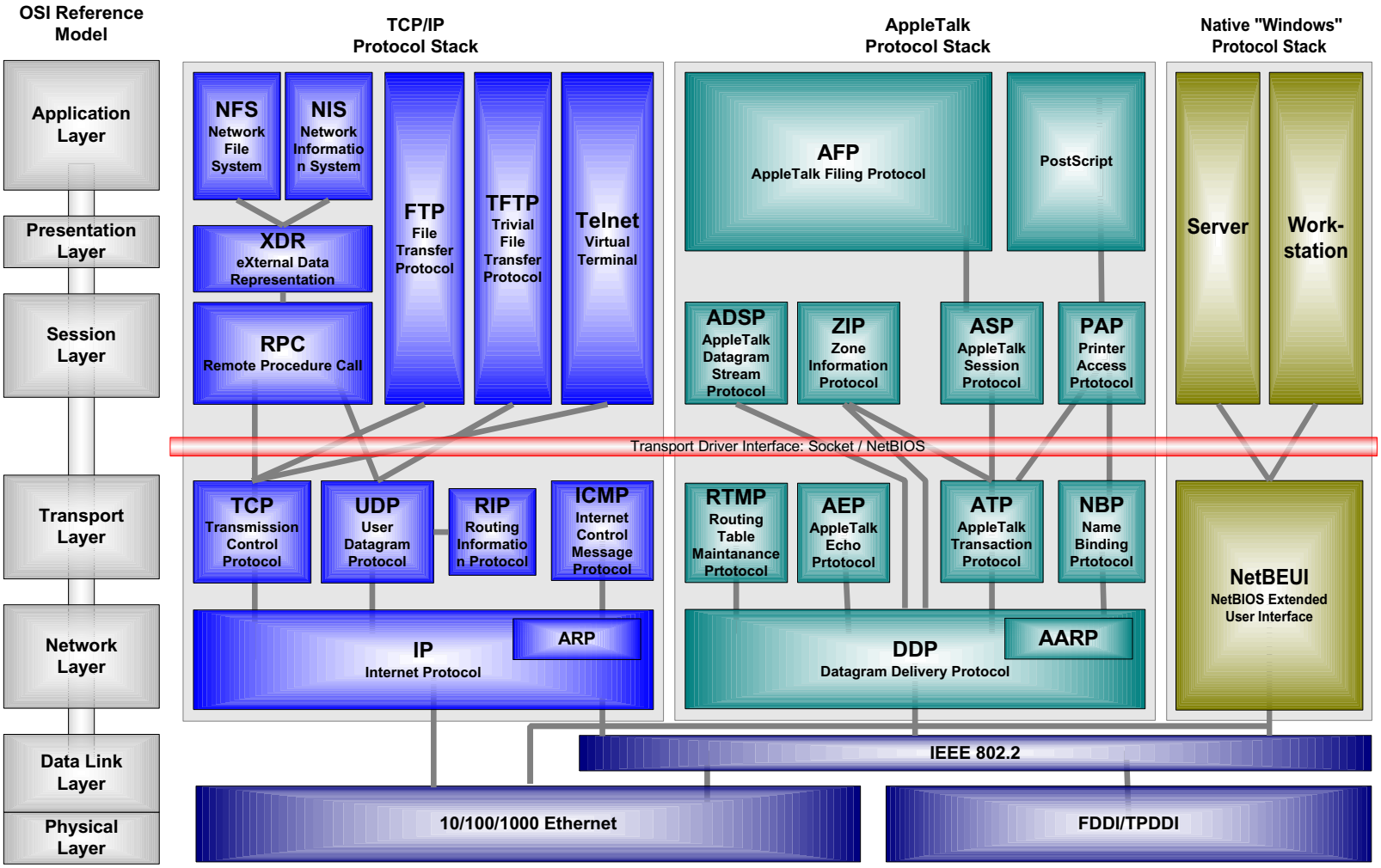
This Open System Interconnection Model was specified by the International Standardisation Organisation (ISO)



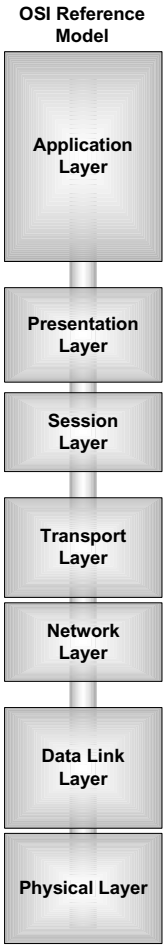
Open System Interconnection 7-Layer Model

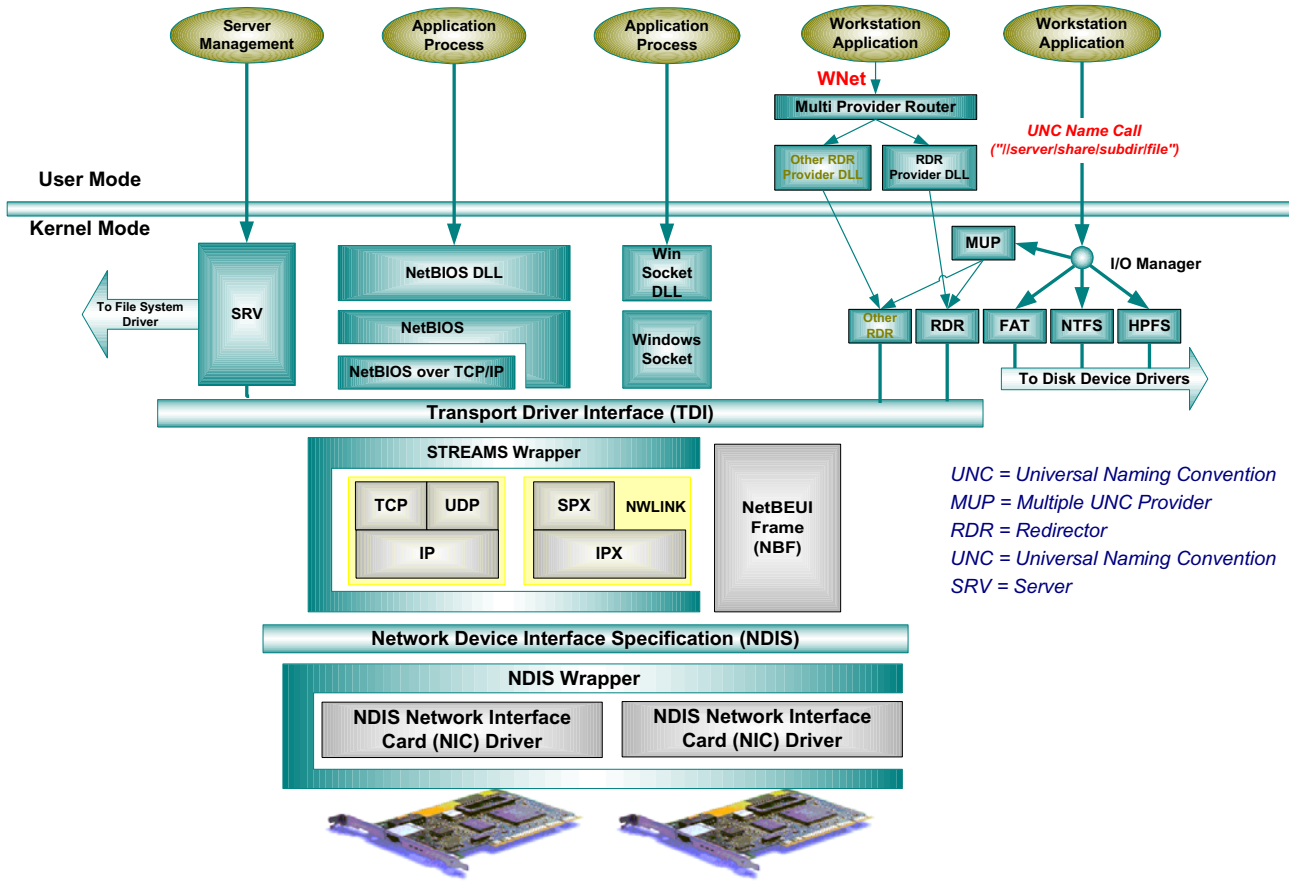
- ***Application Layer:***
  Provides Services to Applications like File Transfer, Electronic Mail, etc.
- ***Presentation Layer:***
  Convertion of data formats to system independent network formats
- ***Session Layer:***
  Set-up, maintenance and release of virtual connections
- ***Transport Layer:***
  End-to-end control of the data transfer with flow control, error recognition/correction, etc.
- ***Network Layer:***
  Routing of data packets within a „Internet“ consisting of sub-networks
- ***Data Link Layer:***
  Creation of transfer frames and control of network media access
- ***Physical Layer:***
  Definition of the physical parameters such as cable type, signal level, transfer code, etc.

Protocol Stacks



Windows NT Network Environment



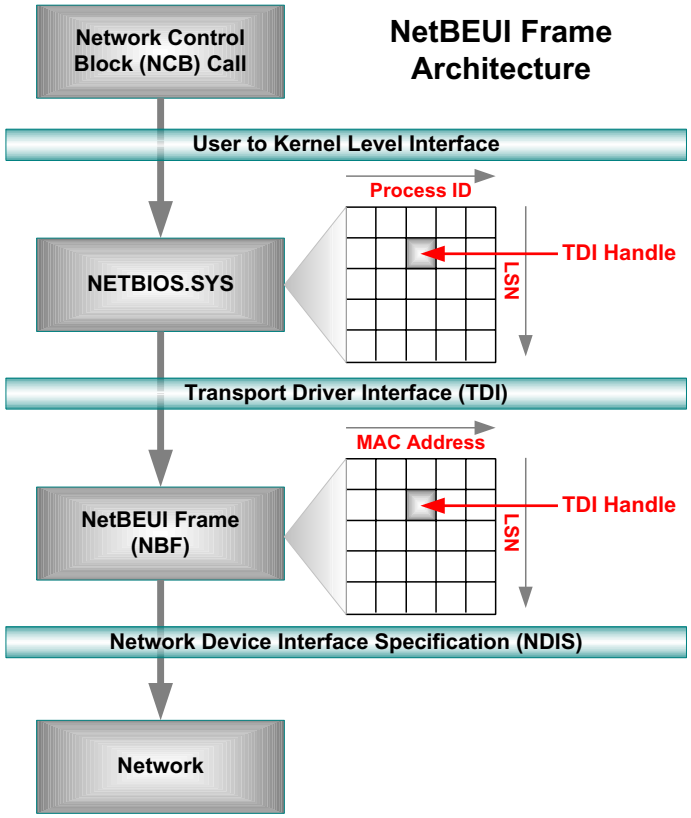


UNC = Universal Naming Convention
MUP = Multiple UNC Provider
RDR = Redirector
UNC = Universal Naming Convention
SRV = Server

Protocol Stacks: NetBEUI

Microsoft NetBEUI

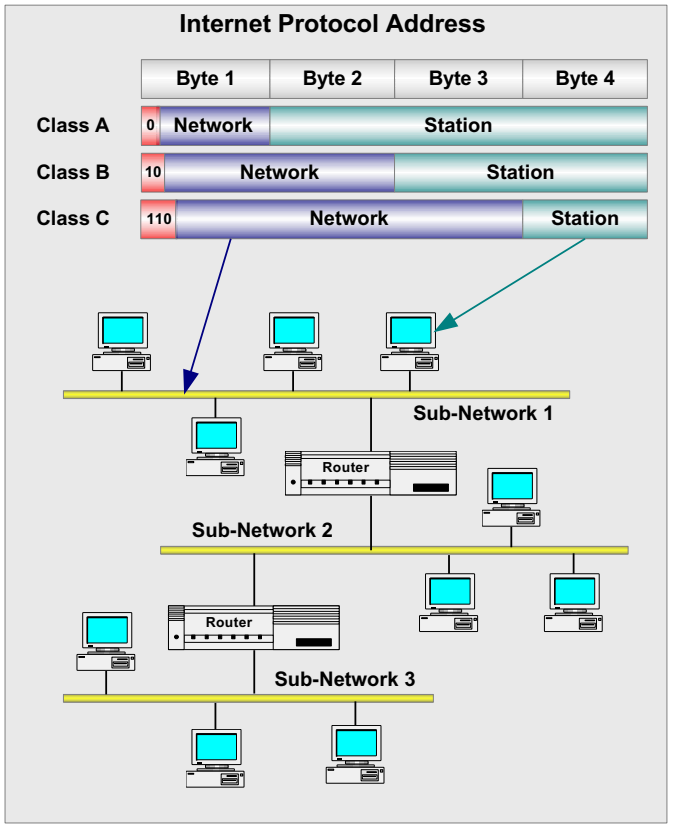
- **NetB**IOS (Network Basic I/O System) **E**xtended **U**ser **I**nterface
- First introduced by IBM in 1985
- Small, efficient, and **non-routable** protocol (NetBEUI traffic can not travel across network boundaries)
- Originally designed for departemental LANs of up to 254 hosts.
- Host addressing scheme is based on MAC-Addresses (i.e. 6 byte Ethernet Addresses), unique logical host and domain names, and **L**ocal **S**ession **N**umbers (**LSN**s)
- An advanded version called **N**et**B**EUI **F**rame (**NBF**) is included with Windowd NT for backward compatibility to LAN Manager and Microsoft-LAN environments. NBF allows much more than 254 host per LAN because of **per Process LSNs**
- Is installed as a default protocol for Windows NT



Protocol Stacks: TCP/IP

TCP/IP

- **TCP/IP**
  Transmission **C**ontrol **P**rotocol / **I**nternet **P**rotocol
- Originally devised by the Defense Advanced Research Projects Agency (DARPA)
- Developed by the **IETF** (Internet Engineering Task Force) and specified in so called RFC (request for comment) documents.
- THE standard network and transport layer protocol
- Uses 4 byte logical **IP-Addresses** to specify nodes in an „Internet“ built by a number of sub-networks.
- An IP-Address consists of a Network Part and a Station Part and is independent of the physical (MAC-) Address of the station
- Is besides NetBEUI and Novell IPX/SPX a standard Protocol Stack under Windows NT



Protocol Stacks: IP-Addresses

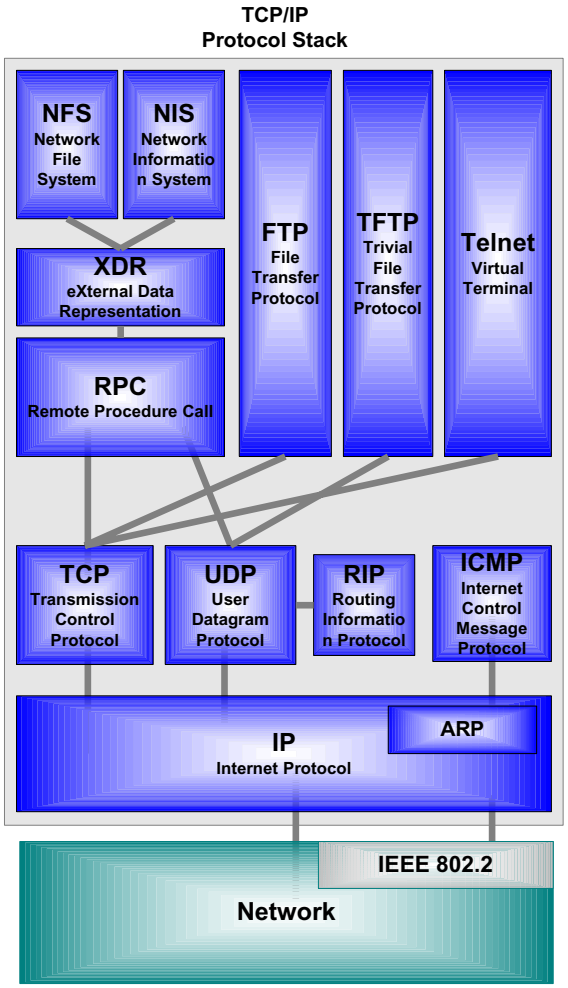
IP Address Details

- A 32 bit IP Address is divided into 4 bytes, each of which is separated by a period (for example, 193.103.71.32). Internet addresses can be decimal, hexadecimal or octal.
- Each of the 4 bytes in the Internet address is automatically interpreted as a decimal, if it is not preceded by a leading zero. If an address byte begins with a zero it is assumed to be octal, if the leading zero is not followed by an ”x” or an ”X”. When followed by an ”x” or ”X” the address is assumed to be hexadecimal. Internet addresses are usually in decimal format; however, each one of the three available formats (decimal, hexadecimal, and octal) is valid.
- Three classes of individual IP Addresses have been defined:
  - **Class A Addresses:** The first byte of a class A address must be a decimal value less than 128. The first byte specifies the (sub-) network and the last three bytes specify a station within the network. The network address 127 is reserved for the local loopback interface and must not be used. The address 92.0.0.10 is an example of a class A Internet number.It defines node 10 (0.0.10) on network number 92.
  - **Class B Addresses:** The first byte of a class B address must be a decimal value greater than or equal to 128, but less than 192. The first two bytes specify the (sub-) network and the remaining two bytes the node within the network. The number 180.138.163.24 is an example of a class B Internet number. It defines node 163.24 on network number 180.138.
  - **Class C Addresses:** The first byte of a class C address must be a decimal value greater than or equal to 192, but less than 224. The first three bytes specify the (sub-) network and the last byte specifies the node within the network. The number 193.103.72.54 is an example of a class C Internet address. It defines node 54 on network number 193.103.72.

Protocol Stacks: TCP/IP

The most essential TCP/IP protocols

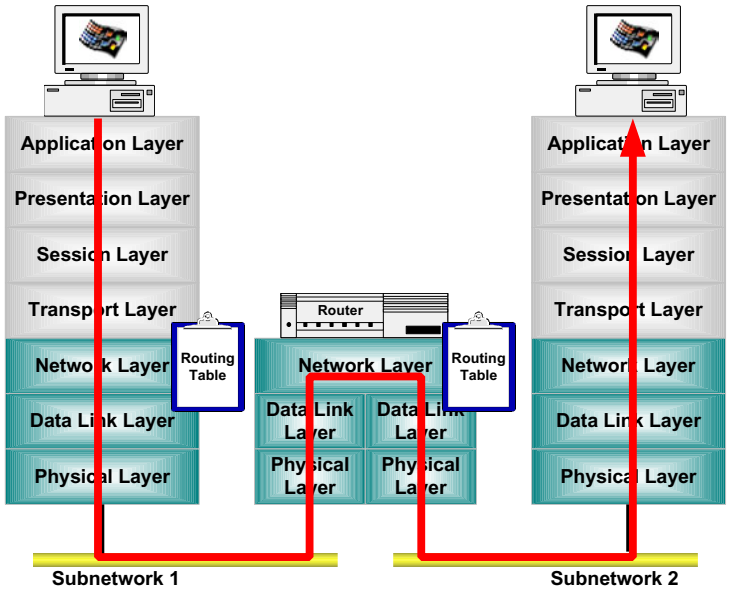
- **TCP (Transmission Control Protocol)** is the base protocol of the Internet protocol family and provides a secured end-to-end controlled, connection-oriented, full duplex data transmission service on the Transport Layer of the OSI 7 Layer Reference Model. TCP uses the IP Internet Protocol to transmit data packets.
- **UDP (User Datagram Protocol)** is a further transmission protocol for the Transport Layer. It provides a connectionless, i.e. unsecured and unacknowledged, data transfer service. Like TCP, UDP also uses the IP Internet protocol to transfer data packets. In contrast to TCP, UDP can also be used to broadcast messages.
- **IP (Internet Protocol)** implements the Network Layer of the OSI Reference Model and employs a datagram service (unacknowledged communication) for unique addressing of nodes and transmission of data packets in an Internet environment. The Internet Protocol addresses target nodes by means of Internet Addresses. Each individual node, such as PCs, workstations or servers in a TCP/IP based network, must have a unique Internet Address



Protocol Stacks: IP Routing

Internet Protocol Routing

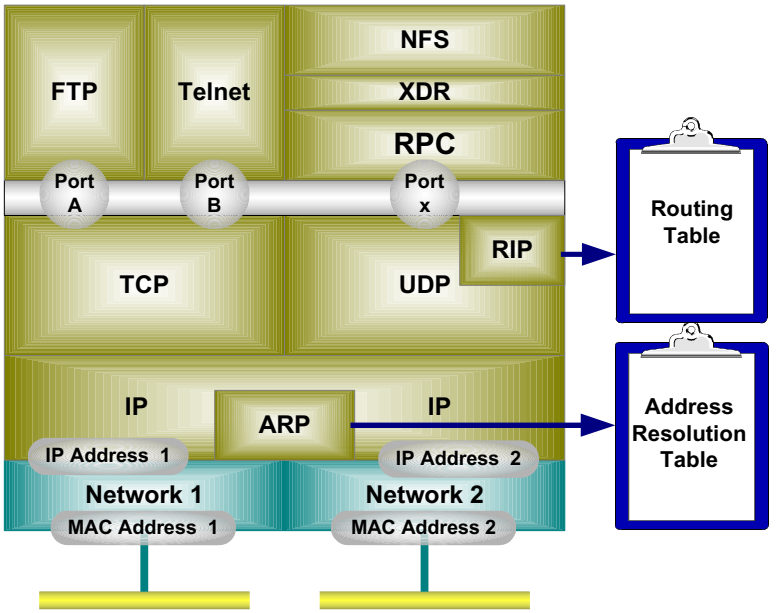
- IP data packets are automatically forwarded between the subnetworks by certain nodes referred to as "**Routers**"
- To allow for correct forwarding of IP packets, each router and also almost each regular station uses its own information table, the **Routing Table**
- This table contains information on whether a target network or node is available via direct „local" access, or whether a router ís needed to forward data packets to another subnetwork
- Routing tables can be edited manually by entering the required information. However, a route path finder protocol is normally employed to dynamically update the routing tables. The protocol most commonly used in IP networks at present is the **RIP** (**Routing Information Protocol**).
- RIP is a so-called Interior Gateway Protocol (**IGP**) by means of which routers make themselves known to all subnetworks to which they have access by transmitting their routing tables to all the nodes attached to the subnetworks.



Protocol Stacks: TCP and UDP Ports

Addressing an Application

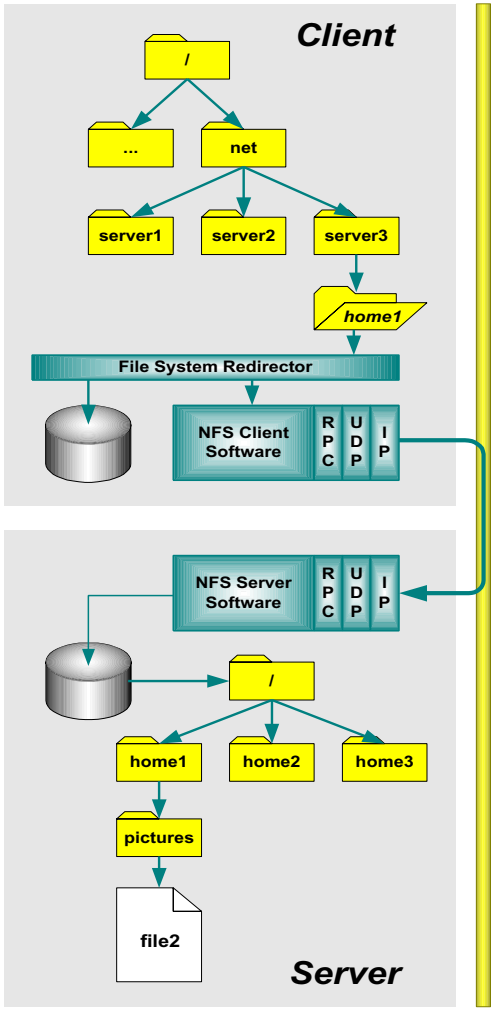
- Host are addressed by 48-bit **MAC Addresses** within a broadcast domain
- Hosts are logically addressed by 32-bit **IP Addresses** within and across broadcast domain boundaries
- IP Addresses are mapped to MAC Adresses using an Address Resolution Table and the **A**ddress **R**esolution **P**rotocol (**ARP**)
- Applications (client and/or server programs) within a machine are addressed via 16-bit Port Identifications (**Port ID**)
- The combination of an IP-Address and a Port-ID identifies a specific application within a specific host and is called a **Socket**
- A combination of a local and a remote Socket uniquely specifies a **Virtual Connection** between applications in different hosts on a network.



Protocol Stacks: NFS

Sun Microsystems Network File System

- **NFS** is a distributed networking file system developed by Sun Microsystems
- Allows servers on a network to share their local volumes with client stations (**NFS Server**)
- Allows client computers to "mount" ("add" to local file sytems) filesystems exported by NFS servers an to access these file systems and their files transparently just like local resources (**NFS Client**).
- Employs a technolkogy known as "**Remote Procedure Call**" (**RPC**). This is a commonly used method of implementing network wide execution of software processes. There are numerous different RPC implementations of which Sun Microsystems' RPC is used by NFS.
- To allow for working with uniform, non-proprietary data structures, RPC uses only data types that comply with "**eXternal Data Representation**" (**XDR**).
- NFS Version 2 and Version 3 were adopted by the **I**nternet **E**ngineering **T**ask **F**orce (**IETF**) and documented in so-called RFC documents.



Protocol Stacks: NFS

NFS Versions

- In the past NFS Version 2 was widely used
- Today most products support NFS Version 3 but also NFS Version 2 for backward compatibility.
- The main differences between V2 and V3 are:
  - **NFS V2**
    - was primarily designed to run on top of UDP/IP
    - uses 32 bit data types and a maximum read/write block size of 8 Kbytes
    - specification only allowed synchronous writes
  - **NFS V3**
    - is designed to run on UDP and as well on TCP.
    - uses 64 bit data types and maximum blocksizes up to 64 KBytes
    - allows synchronous (safe) writes as well as asynchronous (fast) writes.
  - **NFS V3 transfers are faster than NFS V2 ones** because of the larger block sizes and the asynchronous write mode

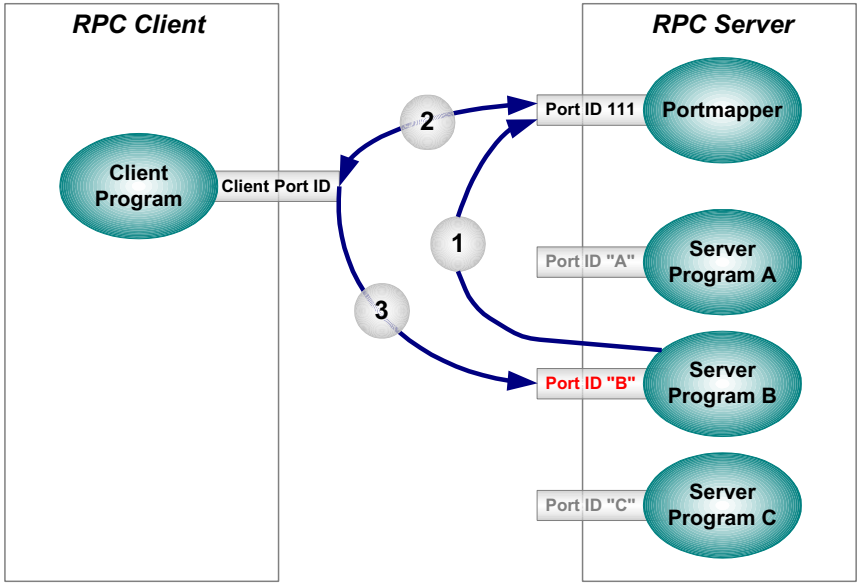
NFS for Windows NT

- Originally NFS was **desinged for UNIX** system environments
- For connectivity to UNIX systems, NFS is also used on DOS, Windows, Windows 95, and Windows NT systems
- Currently NFS/RPC is not supported by Microsoft.
- ***DiskAccess*** from Intergraph and ***Maestro NFS Client*** from Hummingbird Communications Ltd. are examples for Windows 97/ Windows NT NFS Client products.
- ***DiskShare*** from Intergraph and ***Maestro NFS Server*** from Hummingbird Communications Ltd. are examples for Windows 97/ Windows NT NFS Server products.
- Heidelberg Prepress proposes to use the Intergraph NFS products for Windows NT
- Intergraph has licensed the DiskAccess and DiskShare sources to Microsoft. Microsoft will incorporate the Intergraph NFS into its „**Windows NT Services for UNIX**“

Protocol Stacks: The Portmapper

The Portmapper

- The Portmapper is responsible for the **registration and administration** of local services which are made available locally or network wide via Sun **RPC**
- RPC **server processes must register** at the local Portmapper for the assignment of a (UDP or TCP) Port-ID (logical access point to a software process).
- RPC **client processes need a way to look up the port numbers of RPC server processes**. They do this by sending requests to the local or remote Portmapper who then responds with the Port-ID of the requested server process.
- Sun RPC and the Portmapper are **currently not supported by Microsoft** for Windows NT
- **Every third party product** for Windows NT based on Sun RPC **comes with ist own Portmapper** implementation.
- When two or more third party products (e.g. Intergraph DiskShare, Legato Networker, ArrayGUIde/CLARiiON Agent) are installed then the **different Portmapper implementations collide.**

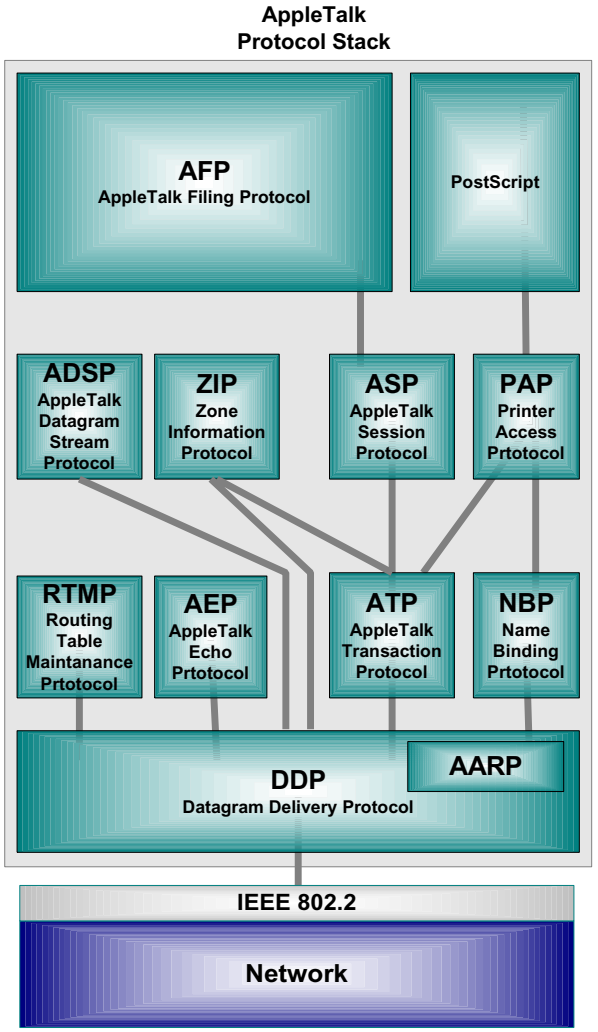


1. **The server program registers with the Portmapper who then assigns a Port ID to the program**
2. **The client program contacts the portmapper for the Port ID of the Server Program. The Portmapper replies with the correct Port ID**
3. **The client program calls the server program via the Port ID it got from the Portmapper**

Protocol Stacks: AppleTalk

AppleTalk

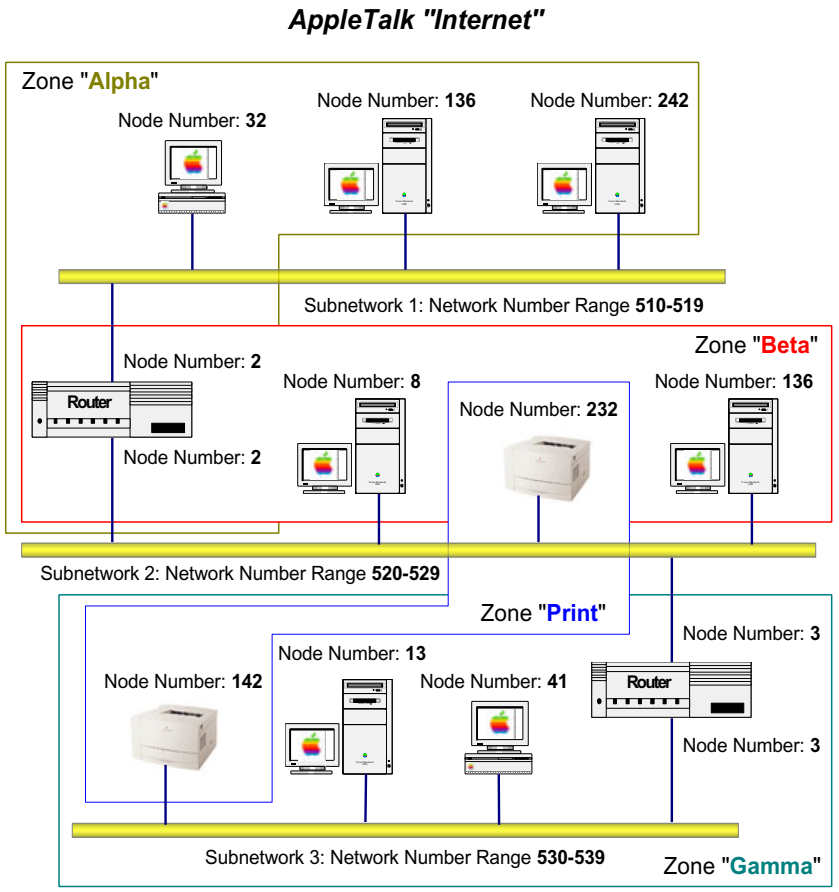
- **AppleTalk** is proprietary protocol family from Apple Computer Inc. which was originally designed for the Apple **LocalTalk** network.
- It has a similar function to the TCP/IP protocol family but is totally incompatible with TCP/IP.
- It implements a File- and a Print-Service
- Today, AppleTalk is commonly implemented on an Ethernet (10/100/1000) basis (**EtherTalk**)
- The AppleTalk protocol family consists of
  - transmission-oriented protocols
    - Datagram Delivery Protocol = **DDP**
    - AppleTalk Transaction Protocol = **ATP**
  - auxiliary protocols
    - AppleTalk Address Resolution Protocol = **AARP**
    - AppleTalk Echo Protocol = **AEP**, etc.
  - application protocols
    - AppleTalk Session Protocol = **ASP**
    - AppleTalk Filing Protocol = **AFP**
    - Printer Access Protocol = **PAP**
- The „**Services for Macintosh**“ for Windows NT implement AppleTalk file and print services



Protocol Stacks: AppleTalk

AppleTalk Addressing Scheme

- The AppleTalk Addressing Scheme within an **AppleTalk „Internet“** is based on:
- **Node Numbers** specify a host within a (physical) network. AppleTalk only allows 254 Node Numbers per Network Number
- **Network Numbers** specify a network within an AppleTalk „Internet“.
  - To allow more than 254 hosts per subnetwork AppleTalk allows to assign a Range of numbers to each subnetwork.
  - A Network Number can be in the range from 1 to 65,279.
- **Zones** are logical groups of hosts to allow a simplified browsing for resourses like file servers or printers.
  - Each host on the network is assignd to a Zone.
  - Each subnetwork can have multiple Zones and a zone can exist in more than one network.
  - The zones are summarized in zone lists.
  - One of the Zones on a network is the default Zone for all hosts on that network



Protocol Stacks: AppleTalk

AppleTalk Router

- AppleTalk packets are forwarded across network boundaries by Routers
- AppleTalk packets can travel across a maximum of 15 router hops
- AppleTalk Routers access the network to which they are connectet via "**Ports**"
  - Ports are typically network adapters and each has a network number (or range) and a zone name (or list)
- A special type of Routers are „**Seed Routers**"
  - Seed Routers initialize and broadcast routing information like network numbers and zone lists (configured by the administrator)
  - Not every Router in an AppleTalk „Internet" must be a Seed Router but each subnetwork must have at least one Seed Router
  - If there are more than one Seed Routers attached to the same subnetwork then they must seed identical information
  - A Seed Router must be started prior to all other nodes within a subnetwork

AppleTalk Frame Size

- AppleTalk runs on Ethernet (10/100/1000) or FDDI networks with an (uneffective) fixed maximum data packet size (**MTU** size = **M**edia **T**ransfer **U**nit size) of **586** bytes.
- The maximum Ethernet MTU size is **1500** bytes and the maximum FDDI MTU size is **4500** bytes. The maximum packet frames are therefore only partly exploited by AppleTalk packets.
- This rigid limit has an obvious negative influence on the achievable data transfer rate between two AppleTalk nodes.
- **AppleShare IP** is a modified AppleTalk Protocol Stack based on TCP/IP instead of ASP/DDP.
  - AppleShare IP is supported for MacOS 8 and higher
  - AppleShare IP allows a much higher file transfer performance than traditional AppleShare, since it uses the full MTU size of the underlying network
  - Currently AppleShare IP is not supported for Windows NT

End of the Training Course on Network Protocols for Windows NT