Application Protocols for TCP/IP Administration

BootP, TFTP, DHCP

Agenda

- **BootP**
- **TFTP**
- **DHCP**

BootP (RFC 951, 1542, 2132)

- **BootP was developed to replace RARP**
  - capabilities of RARP (determination of IP-address) plus bootstrap ability
- **bootstrapping**
  - allows diskless clients (and other network components without non-volatile memory) to load operating system code and configuration parameters from a central server
- **BootP is based on a client-server principle and uses UDP communication**
  - client-side: well known port 68
  - server-side: well known port 67

BootP-Principles

- **BootP-client sends request to the BootP-server**
  - using 255.255.255.255 as destination address (limited broadcast)
  - and 0.0.0.0 as source address (UDP relies upon IP!)
- **server uses the client's MAC-address for a database lookup to determine the IP-address of the client**
- **server replies with the desired boot information; again a limited broadcast is used as destination address**
  - alternatively, an ARP-cache entry without utilizing the ARP-request/response-procedure at the server-side
- **end of the BootP-procedure**

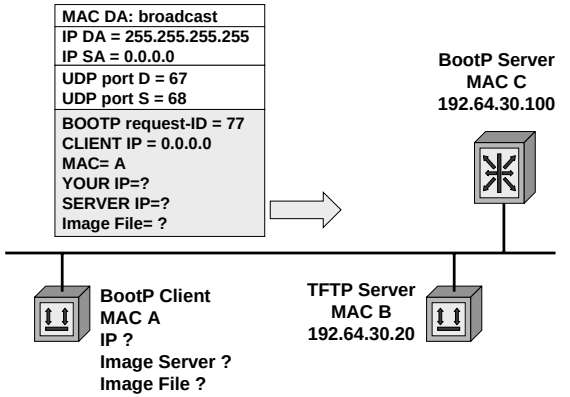
BootP-Principles

- **basically boot information contains**
  - the IP-address of an IP-host which provides appropriate bootfiles (image + configuration)
  - and also the filename of these bootfiles
- **client uses this information to load bootfiles via TFTP**
- **limited broadcast is restricted on a single LAN; in order to reach also BootP-servers of other subnets**
  - router or other computer-system must be designed and configured appropriately to act as BootP-relay agent
    - configuration of an IP-helper-address (Cisco specific) to forward specific UDP broadcasts

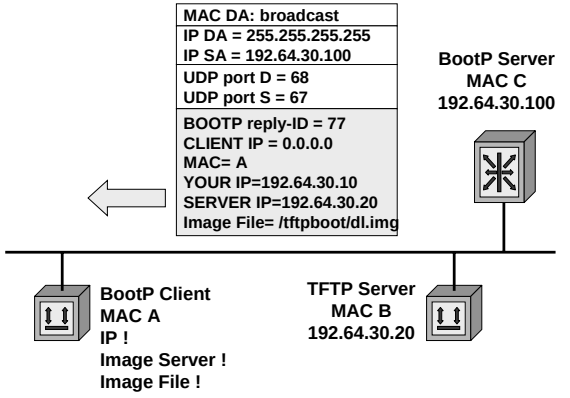
BootP-Principles

- **separation of the boot task into a BootP-part and a TFTP-part means:**
  - BootP-server only needs to maintain a small reference database
  - image- and configuration-files can be stored on another machine
- **the BootP client is responsible for error detection (retransmission after timeout)**
  - UDP and checksum is used for payload as IP's checksum doesn't take the data field into account
  - IP datagram has the "Do Not Fragment-Bit" set to one
  - timeout is selected randomly from a special interval, which is increased as errors last on -> avoiding network overload!

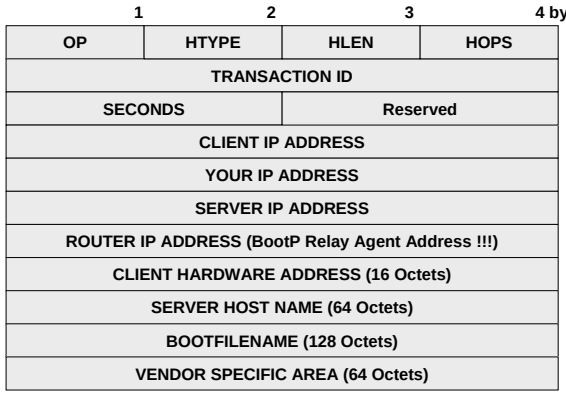
Bootstrap 1



Bootstrap 2



BootP-Message Format



BootP Message Fields

- **OP (Operation Code):**
  - 1 ... Boot Request, 2 ... Boot Reply
- **HTYPE (Hardware Type):**
  - network type (1 for Ethernet); numbers similar to ARP
- **HLEN:**
  - length of the hardware address (e.g. 6 for ethernet)
- **HOPS:**
  - number of hops; optionally used by routers
  - initialized with zero by the client
  - increased by one if a BootP-server forwards the request to other servers (bootstrap over multiple servers)
    - BootP relay agent activated

BootP Message Fields

- **TRANSACTION ID:**
  - identification mark of related request-reply BootP-datagram's (random number)
- **SECONDS:**
  - seconds elapsed since client started trying to boot
- **CLIENT IP ADDRESS:**
  - client IP-address; filled in by client in boot-request if known
- **YOUR IP ADDRESS:**
  - client IP-address; filled in by server if client doesn't know its own address (if the client IP-address in the request was 0.0.0.0)

BootP Message Fields

- **SERVER IP ADDRESS:**
  - server IP-address where image is stored; returned in boot-reply by the server
- **ROUTER IP ADDRESS:**
  - server is part of another subnet
  - IP address of the BootP relay agent
- **CLIENT HARDWARE ADDRESS:**
  - MAC-address of client
  - advantage of BootP in comparison to RARP: server-application may rely upon UDP/IP protocol-stack to extract MAC-address; no need for layer 2 access

BootP Message Fields

- **SERVER HOST NAME:**
  - optional server host name
- **BOOTFILENAME:**
  - contains directory path and filename of the bootfile
- **VENDOR SPECIFIC AREA:**
  - may optionally contain vendor information of the BootP-server
  - according to RFC 2132 it is also possible to mention the subnet-mask (opt. 1), hostname, domain name, IP-address of the DNS-server (opt. 6), IP-address of the default gateway (Router opt. 3), etc.
  - Here DHCP comes in (opt. 53) !!!

Agenda

- **BootP**
- **TFTP**
- **DHCP**

Trivial File Transfer Protocol (RFC 1350)

- **TFTP is suited for applications**
  - that do not require the rather complex procedures of FTP
  - or cannot provide enough resources (RAM, ROM)
- **typical utilization:**
  - boot helper for diskless clients
  - enables software-update for network components like bridges, router, SNMP agents of hubs, etc.
- **code size of TFTP is very small and easy to implement**
  - fits well in Bootstrap-ROMs of workstations

TFTP

- **TFTP has been designed to provide**
  - *simplest* transmission of files
  - client-server communication principle
- **TFTP do NOT support**
  - functions for reading directory contents
  - access verification mechanisms
- **TFTP is an unsecured protocol,**
  - there is no authentication (no username or password)

TFTP

- **TFTP uses UDP**
  - well know port server 69, datagram size = 512 bytes
- **TFTP is responsible for error recovery**
  - based on IdleRQ-protocol (stop and wait)
- **IdleRQ-principle**
  - every TFTP-datagram is marked with a sequence number
  - these datagram's are confirmed by short ACK-datagram's in the opposite direction
  - after receiving an acknowledge the next datagram is send
  - error recovery by retransmission after a timer expires
    - timer is activated after sending data or acknowledges
    - TFTP uses adaptive timeout (e.g. exponential backoff algorithm)

TFTP Message Formats

2 octet opcode	n octets	1 octet	n octets	1 octet
READ REQUEST (1)	FILENAME	0	MODE	0

Type 1

2 octet opcode	n octets	1 octet	n octets	1 octet
WRITE REQUEST (2)	FILENAME	0	MODE	0

Type 2

- **Type 1 and 2 initialize the TFTP transfer by specifying the direction of the transaction of the file**
- **MODE determines the type of data (NETASCII, BINARY, MAIL)**
- **FILENAME and MODE can have arbitrary length and consist of ASCII characters; the last character is always NULL**

TFTP Message Formats

2 octet opcode	2 octet seq.#	up to 512 octets
DATA (3)	BLOCK#	INFORMATION OCTETS

Type 3

2 octet opcode	2 octets
ACK (4)	BLOCK#

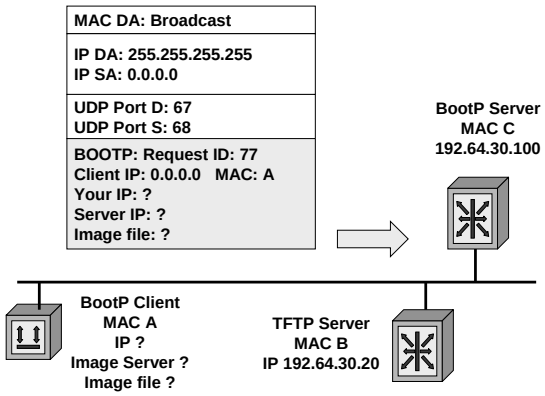
Type 4

- **Type 3 is used for the data transfer**
- **BLOCK# is the sequence number (starting with 1, increased by one for every block)**
- **last block has length < 512 (EOF mark)**
- **Type 4 is used to acknowledge every DATA message explicitly**

TFTP Protocol Description

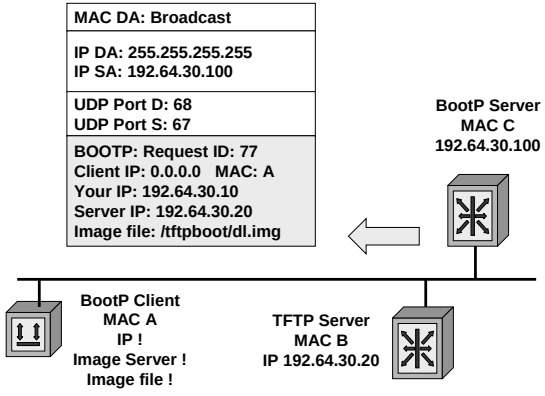
- a TFTP transfer begins with the request to read or write a file
- if the server accepts the request, a connection is opened and datagram's, with a fixed size of 512 bytes, are sent
  - all datagram's are numbered consecutively beginning with 1,2,3,...and so on
  - each datagram must be acknowledged
- the connection will terminate if a datagram arrives with less than 512 bytes, or in case of errors
  - retransmission will start in case of datagram loss

Bootstrap (1)



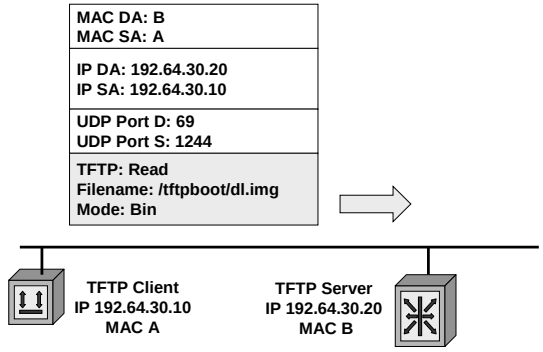
 BootP, TFTP, DHCP, v4.4 21

Bootstrap (2)



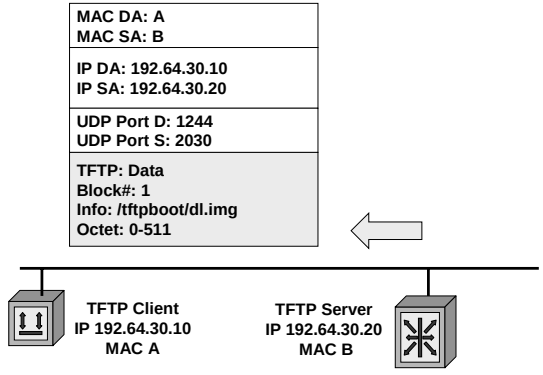
 BootP, TFTP, DHCP, v4.4 22

TFTP (1)



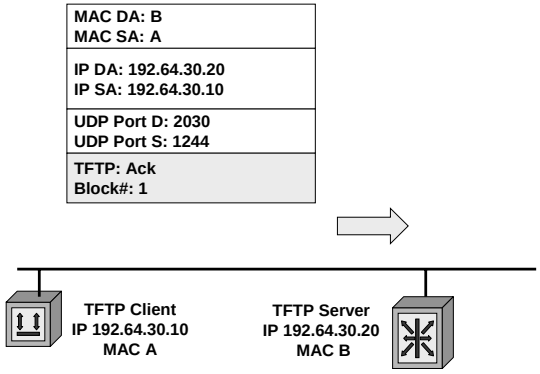
 BootP, TFTP, DHCP, v4.4 23

TFTP (2)

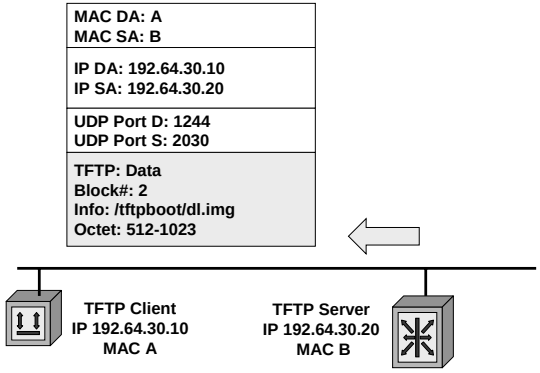


 BootP, TFTP, DHCP, v4.4 24

TFTP (3)



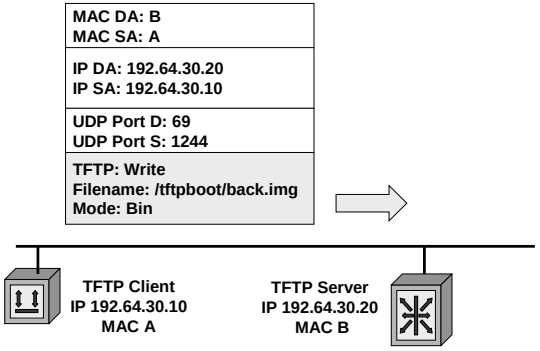
TFTP (4)



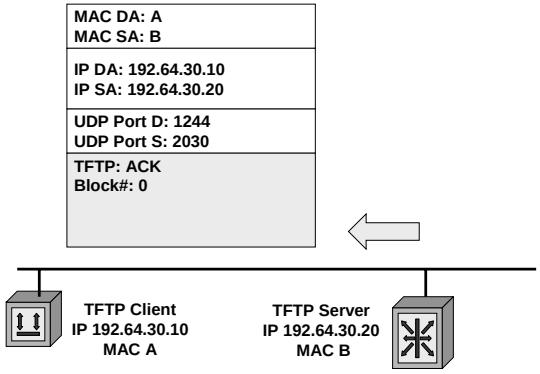
TFTP User Interface

- **Basic TFTP commands:**
  - **Connect** <host>: Destination host
  - **Mode** <ascii/binary>
  - **Get** <remote file> [<local filename>]: Retrieve a file
  - **Put** <remote file> [<local filename>]: Send a file
  - **Verbose** <on/off>: shows status information during the transfer.
  - **Quit**: Exit TFTP
- **TFTP data modes:**
  - **NETASCII**: 8 bit character set.
  - **OCTET**: Binary or 8 bit raw
  - **MAIL**: Allows sending a mail to a user, rather than transferring to a file.

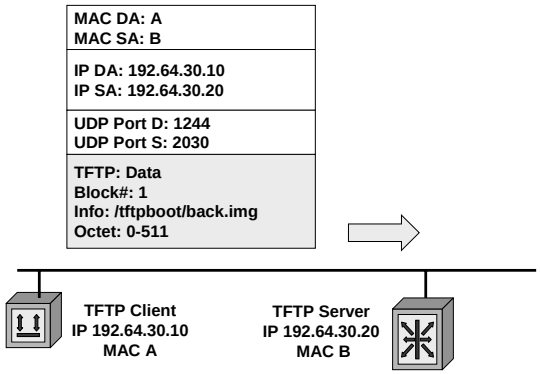
TFTP (1) Write



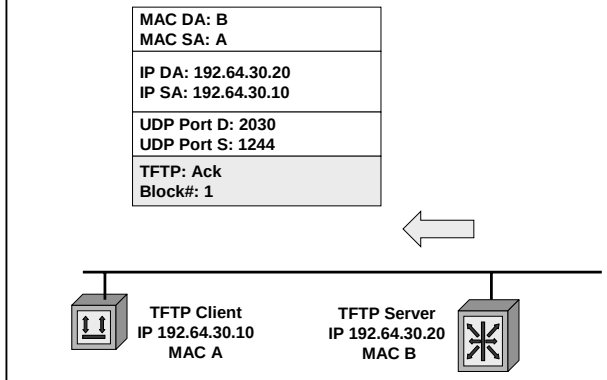
TFTP (2) Write



TFTP (3) Write



TFTP (4) Write



Agenda

- **BootP**
- **TFTP**
- **DHCP**

DHCP (Dynamic Host Configuration Protocol)

- **DHCP (RFC 2131, 3396) build on two components:**
  - Protocol to deliver host specific configurations from a server to its client
  - Mechanism to allocate temporary or permanent host addresses
- **Temporary address allocation**
  - DHCP server receives a request from a DHCP client and picks out an IP address from a configurable address pool and offers this address to the client
  - the client can use this leased address for a period of time
  - after the end of this lease, the address must again be requested by the client or is returned to the address pool

DHCP Configurable Parameters

- **DHCP eliminates**
  - a number of configuration tasks and problems associated with a manual TCP/IP configuration
- **A DHCP client can asks for:**
  - IP address
  - Subnet Mask
  - DNS Server, NetBIOS-Name Server
  - default TTL, Source Routing Option, MTU
  - max. Fragment Size, Broadcast Address
  - List of Default Gateways + Preferences, Static Routes
  - ARP Cache Timeout, TCP Keepalives
  - Ethernet Encapsulation
  - Path MTU Discovery (RFC1191)
  - Router Discovery (RFC 1256)

DHCP Basics

- **DHCP provides a framework for delivering configuration parameters to hosts based on a TCP/IP Network.**
- **DHCP is based on BootP using the options field (opt. 53) of the BootP header in order to:**
  - lookup unused network addresses.
  - support configuration options.
- **DHCP uses port 67 UDP (BootP Server) and port 68 (BootP Client).**

DHCP Basics (cont.)

- **DHCP provides three mechanisms for address allocation:**
  - Automatic:
    - DHCP assigns a permanent address to a host
  - Dynamic:
    - DHCP gives the client an address for a limited time period (LEASE). Automatic reuse of not active addresses is possible.
  - Manual:
    - Host addresses are still manually configured by a Network Administrator but other parameters configured by DHCP

BootP/DHCP Message Format

code	HWtype	length	hops
Transaction ID			
seconds		Flags field	
Client IP address			
Your IP address			
Server IP address			
Router IP address (DHCP Relay Agent Address !!!)			
Client HW Address 64 byte			
Server host name 64 byte			
Boot file name 128 byte			
Options variable length (at least 312 byte) (here are the DHCP messages !!!)			

BootP/DHCP Message Format (cont.)

- **Code:**
  - Indicates Request (1) or Reply (2).
- **HWtype:**
  - Type of hardware, Ethernet (1) IEEE 802 (6).
- **Length:**
  - MAC Address length
- **Hops:**
  - Is set by the client to zero, incremented by Relay Agent who requests to another server and is used to identify loops.

BootP/DHCP Message Format (cont.)

- **Transaction ID:**
  - Random number used to match this boot request with the response it generates.
- **Seconds:**
  - Is the elapsed time in sec. since the client started booting.
- **Flags field:**
  - MSB is used as a broadcast flag. Other bits are set to zero.
- **Client IP address:**
  - Set by the client. Either its known IP address, or 0.0.0.0.
- **Your IP address:**
  - Set by the server, if the clients address is set to 0.0.0.0.

BootP/DHCP Message Format (cont.)

- **Server IP address:**
  - Set by the server
- **Router IP address:**
  - The address of a BOOTP relay agent
- **Client HW address:**
  - Set by the client. DHCP uses special IDs or the MAC address to identify the client
- **Server host name:**
  - Name of the server
- **Boot file name:**
  - Set by the client to zero, or specifies a boot file. In a DHCPDISCOVER also zero, in the DHCPOFFER a full directory path from the server will be returned.

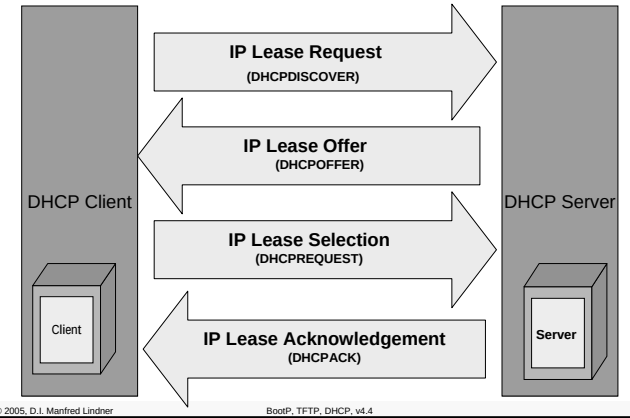
DHCP Message Types in Option Field

- DHCPDISCOVER (opt. 53 / type 1):
  - Client broadcast to find DHCP server(s).
- DHCPOFFER (opt. 53 / type 2):
  - Response to a DHCPDISCOVER, offering an IP address and other parameters.
- DHCPREQUEST (opt. 53 / type 3):
  - Message form the client to the server to get the following:
    - Requests the parameters offered by one server, declines all other offers.
    - Verification of a previously allocated address after a system reboot, or network change.
    - Request the extension of the lease time.

DHCP Message Types (cont.)

- DHCPACK (opt. 53 / type 5):
  - Acknowledgement from server to client, with IP address and parameters.
- DHCPNACK (opt. 53 / type 6):
  - Negative ACK from server to client.
  - Clients lease expired or requested IP address is invalid.
- DHCPDECLINE (opt. 53 / type 4):
  - Message from a client to a server indicating an error.
- DHCPRELEASE (opt. 53 / type 7):
  - Message from a client to a server cancelling remainder of a lease and relinquishing network address.
- DHCPINFORM (opt. 53 / type 8):
  - Message from a client that has already an externally configured IP address, asking for more local configuration parameters

DHCP Operation



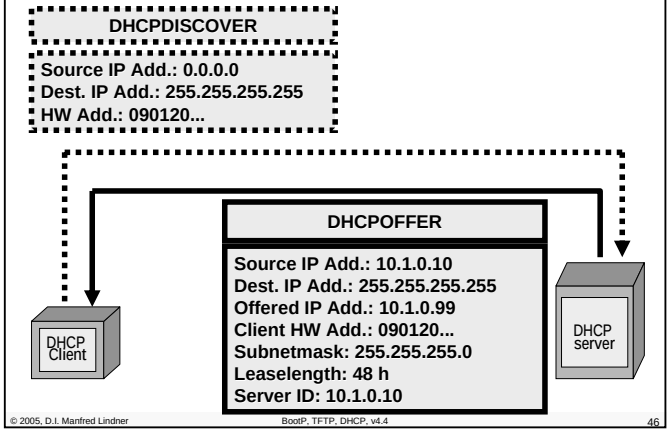
IP Lease Request

- **When the clients starts up**
  - sends a broadcast to all DHCP servers
  - Since the client has no IP configuration, it uses 0.0.0.0 as source- and 255.255.255.255 destination address
  - This request is send in a DHCPDISCOVER message, together with the clients HW- address and the computer name
- **The IP lease is used when:**
  - TCP/IP initializes for the first time on this client
  - The client requests a specific IP address and is denied
  - The client previously leased an IP address, but released the lease and requires a new lease

IP Lease Offer

- **All DHCP servers**
  - that receive the DHCPDISCOVER message and has valid IP information for this client
  - send out a DHCPOFFER (broadcast) that includes:
    - Clients HW address
    - An offered IP address (in the Your IP Address Field)
    - Subnet Mask (in the Options Field)
    - Length of the lease (time value)
    - Server ID or the IP address of the offering DHCP server

IP Lease and Offer



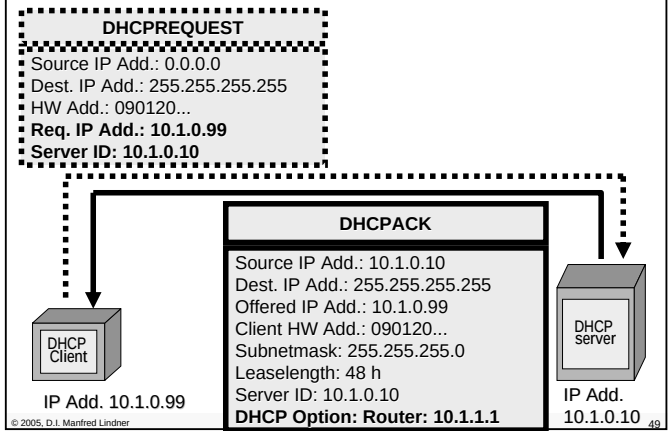
IP Lease Selection

- **When a client receives**
  - an offer from at least one DHCP server
  - he sends a DHCPREQUEST (broadcast) out to the network, to tell all the other DHCP server that no more offers are accepted
  - the DHCPREQUEST message includes the server ID (IP address) of the server whose offer was accepted by the client

IP Lease ACK / NACK

- **In case of success a DHCPACK is send by the server whose offer was accepted**
  - DHCPACK contains a valid lease for an IP address and possible other configuration parameters
  - After the client receives the DHCPACK, TCP/IP is completely initialized and the client enters the BOUND state
  - If the client is bound, it can use TCP/IP as a base for communication
- **In case of no success a DHCPNACK will be send:**
  - e.g. Client tries to lease the previous IP address, but this address is no longer available
  - e.g. Client's IP address is invalid, the client may have been moved to an other subnet

IP Lease Selection and ACK



DHCP Lease Renew

- **When the server sends his DHCPACK**
  - containing the IP address for the client, the beginning of the lease period is registered
- **The lease time is located**
  - in the DHCPACK message in addition to two other time values T1 and T2
- **T1 (Renewal Attempt) and T2 (Sub Renewal Attempt)**
  - are configured at the DHCP server.
  - T1= 0,5 x lease time, T2= 0,875 x lease time.

DHCP Lease Renew (cont.)

- **T1 and T2 start their function**
  - when the client is bound.
  - The client attempt to renew the lease when 0,5 of the lease time has expired.
  - The client enters the RENEWING state and sends an DHCPREQUEST (unicast) to the server forcing him to extend the lease.
  - If the server accepts, an DHCPACK, containing a new lease time and the default values of T1/T2 are sent back to the client.

DHCP Lease Renew (cont.)

- **If the lease could not be renewed**
  - at the 0,5 interval, the client will contact any other DHCP server DHCPREQUEST (using broadcast) when 0,875 of the lease time has expired to renew the clients lease time.
- **The client enters the REBINDING state**
  - when 0,875 of the lease time has expired .
- **Any DHCP server can answer to this request**
  - with an DHCPACK renewing the lease, or with an DHCPNACK, forcing the client to reinitialize and to get a new lease for an other IP address.

DHCP Lease Renew (cont.)

- **Generally:**
  - If a lease expires or an DHCPNACK is received, the client must stop using it's present IP address.
  - This will result in TCP/IP communication stop for this client.
  - The client must request a new lease using DHCPDISCOVER.

DHCP over Subnets

- **Note that:**
  - DHCP is related to BOOTP.
  - DHCP messages are broadcast based (L2-Ethernet-Broadcast and IP-Limited Broadcast), so they can not be forwarded by a router.
  - In case of connecting DHCP clients to their servers over a number of subnets which are connected with routers, it is unavoidable to enable the broadcast forwarding on this router = BOOTP relay agent.
  - Most of the routers support this specific function.
  - On a router, broadcast forwarding is turned OFF by default.

DHCP Considerations

- **What IP address options will the client obtain from the DHCP server?**
  - Default Gateway, DNS, WINS, Relay Agent.
- **Which Computers will become DHCP clients?**
  - Non DHCP clients will have static IP addresses, like routers, servers, management-stations.
  - Static IP addresses must excluded from the DHCP address pool.
- **DHCP over multiple subnets?**
  - Configure router as BootP relay agents or install a DHCP server for each subnet.
  - Note: DHCP servers don't share address information, like WINS does, so in case of multiple DHCP servers, create DHCP server groups containing their member clients