| BOOTP - Bootstrap Protocol (BOOTP Client and Server in same subnet)							
subnet 1				intranet	subnet 2		**EventHelix.com/EventStudio 1.0**
client		server 1		router	server 2		
RAM software	ROM software	BOOTP Server 1	TFTP Server 1	BOOTP Relay Agent	BOOTP Server 2	TFTP Server 2	10-Feb-02 23:08 (Page 1)

The Bootstrap Protocol (BOOTP) enables a host to boot from ROM and request it's own IP address, a gateway address and a boot file name. The boot file is used to load the disk image into RAM.

Determine Hardware Address

Client obtains its own hardware address from the ROM

LEG: BOOTP within same subnet

In this scenario we cover the case where the BOOTP Server is directly connected to the BOOTP Client

BOOTP_Request
hw_addr, UDP_Port = N67, src_ip = zero, dst_ip = broadcast
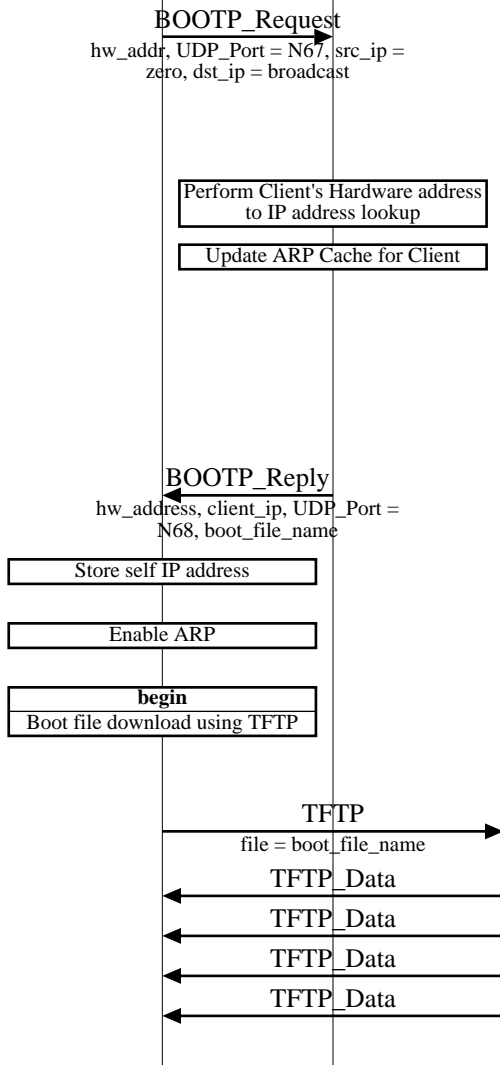
Client sends a UDP datagram with a BOOTP_Request. The destination UDP port number is 67. Since the client does not know its own IP address it sets the source IP address to 0.0.0.0. In many cases the client does not know the IP address of the server, so it sets the destination IP address to broadcast (255.255.255.255).

Perform Client's Hardware address to IP address lookup

Server looks up the client's hardware address in a configuration file to determine the IP address to be assigned to the client

Update ARP Cache for Client

The server needs to send an IP message to the client. This cannot be done directly as the ARP cache on the server does not contain IP address to hardware address mapping for the client. The default action of using ARP to determine the hardware address will not work as the client does not know its IP address. This issue is resolved by directly updating the ARP cache with an IP address to hardware address mapping for the client

BOOTP_Reply
hw_address, client_ip, UDP_Port = N68, boot_file_name

The Server sends out the BOOTP Reply using the client's hardware address. The reply is addressed to UDP Port 68. The boot file name will also be sent out to the client.

Store self IP address

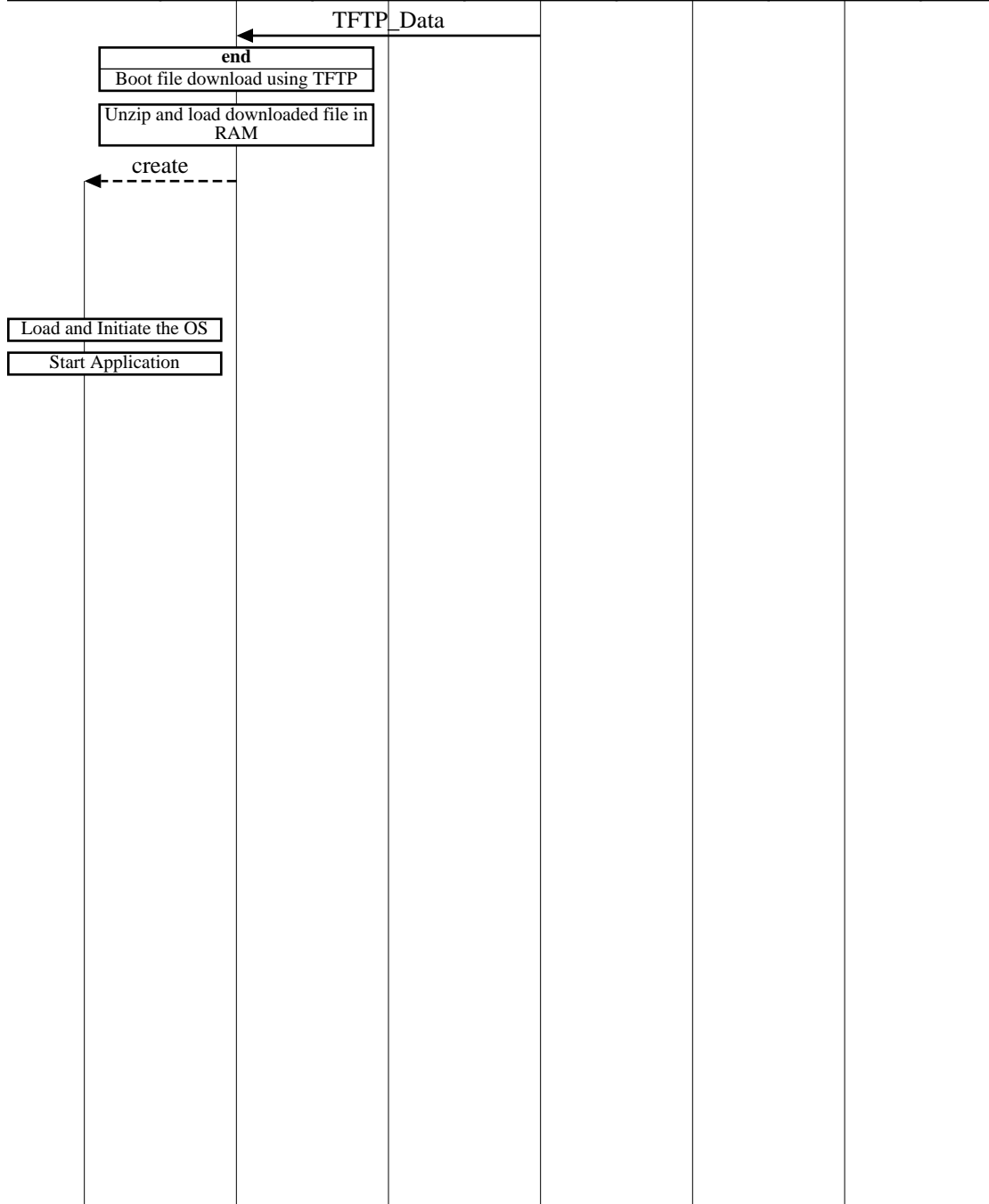Client stores its own IP address. This IP address will be used in all communications with the outside world

Enable ARP

Since the client has received its IP address, it can participate in ARP

begin
Boot file download using TFTP

Now the client can initiate boot file loading. Typically TFTP is used in downloading the file. TFTP is selected over FTP as it uses UDP. This minimizes the amount of protocol code that needs to be burst into the client ROM.

TFTP
file = boot_file_name

Boot file name specified in BOOTP Reply is used to initiate TFTP

TFTP_Data

Complete contents of the file are loaded

TFTP_Data

TFTP_Data

TFTP_Data

BOOTP - Bootstrap Protocol (BOOTP Client and Server in same subnet)

	subnet 1			intranet	subnet 2		EventHelix.com/EventStudio 1.0
	client		server 1	router	server 2		
RAM software	ROM software	BOOTP Server 1	TFTP Server 1	BOOTP Relay Agent	BOOTP Server 2	TFTP Server 2	10-Feb-02 23:08 (Page 2)

TFTP_Data

end

Boot file download using TFTP

TFTP session has ended. Now control will be transferred to the software that has just been downloaded.

Unzip and load downloaded file in RAM

The downloaded file is uncompressed and loaded into RAM

create

ROM software transfers control to the RAM software. The downloaded software includes the OS as well as the application, so no other downloads are required

Note that the ROM software will include a complete implementation of the TCP/IP stack that would replace the primitive stack in the ROM.

Load and Initiate the OS

The Downloaded OS is booted

Start Application

The Downloaded application is started

|						
BOOTP - Bootstrap Protocol (BOOTP forwarding via a BOOTP Relay Agent)						
subnet 1				intranet	subnet 2	**EventHelix.com/EventStudio 1.0**
client		server 1		router	server 2	
RAM software	ROM software	BOOTP Server 1	TFTP Server 1	BOOTP Relay Agent	BOOTP Server 2 / TFTP Server 2	10-Feb-02 23:08 (Page 3)

The Bootstrap Protocol (BOOTP) enables a host to boot from ROM and request it's own IP address, a gateway address and a boot file name. The boot file is used to load the disk image into RAM.

Determine Hardware Address

Client obtains its own hardware address from the ROM

LEG: BOOTP across subnets - Using the BOOTP Relay Agent

In this scenario we cover the case where the BOOTP Server is not directly connected to the BOOTP Client. Here a BOOTP Relay Agent is used to connect the BOOTP Client to BOOTP Server. (Server1 is assumed to be down in this scenario)

BOOTP_Request
hw_addr, UDP_Port = N67, src_ip = zero, dst_ip = broadcast, hop_count = ZERO
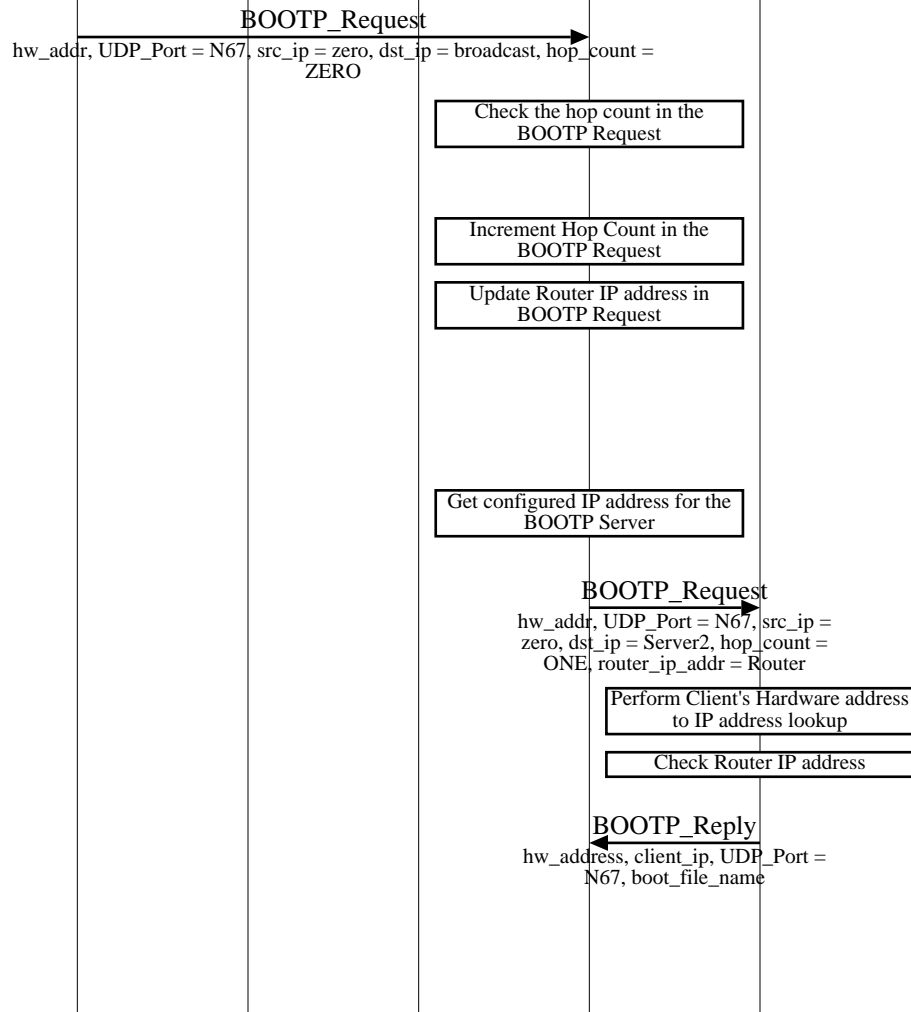
Client sends out a BOOTP request as a broadcast. If there is no BOOTP server present on the subnet, a router (configured as a BOOTP Relay Agent) receives the BOOTP Request

Check the hop count in the BOOTP Request

The BOOTP Relay Agent checks if the hop count in the BOOTP Request is less than a preconfigured threshold. In this case hop count is 0, so the relay agent decides to forward the BOOTP request

Increment Hop Count in the BOOTP Request

BOOTP relay agent decides to forward the message so it increments the hop count in the message

Update Router IP address in BOOTP Request

Client sends a BOOTP Request with a 0.0.0.0 Router IP address. Since a BOOTP relay agent is routing the message, it updates the Router IP address in the message to its own IP address.

Note: BOOTP Relay Agent does not update the router IP address in the message if another relay agent's address is already present

Get configured IP address for the BOOTP Server

BOOTP relay agent determines the BOOTP servers IP address. This IP address has been configured by the network administrator

BOOTP_Request
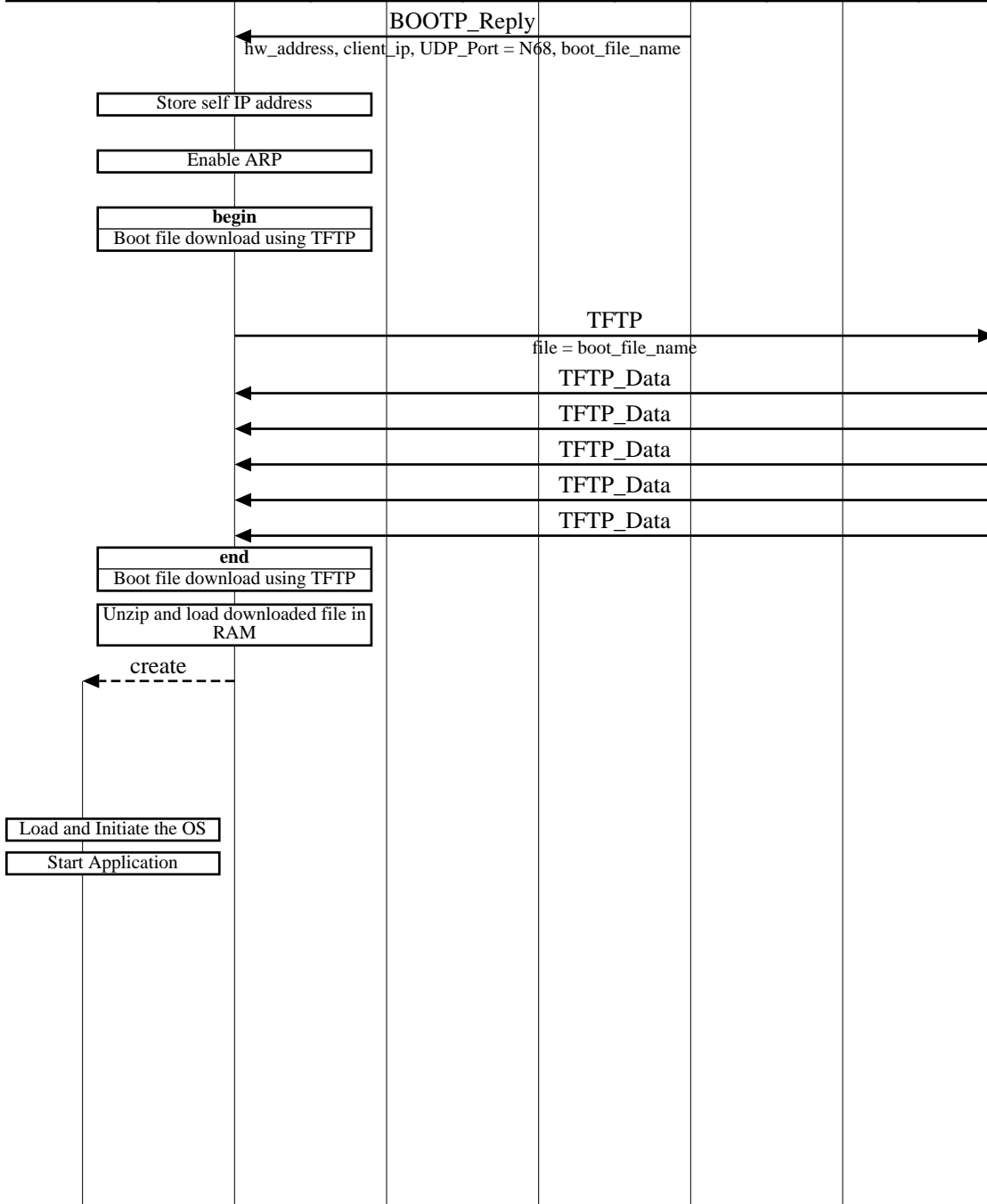hw_addr, UDP_Port = N67, src_ip = zero, dst_ip = Server2, hop_count = ONE, router_ip_addr = Router

BOOTP Relay agent forwards the BOOTP Request as a unicast to the BOOTP Server. There is no need to use a broadcast as the relay agent knows the BOOTP server's IP address

Perform Client's Hardware address to IP address lookup

Server looks up the client's hardware address in a configuration file to determine the IP address to be assigned to the client

Check Router IP address

BOOTP server checks if the BOOTP Request contains a non zero router IP address

BOOTP_Reply
hw_address, client_ip, UDP_Port = N67, boot_file_name

Since the Router IP address is non zero, the BOOTP Reply is sent as a unicast to the router IP address in the message. This message always gets forwarded (using UDP Port 67) to the BOOTP Relay Agent that is directly connected to the client. (As BOOTP Relay Agents forwarding the BOOTP Request from another relay agent do NOT update the router IP address)

subnet 1			intranet	subnet 2		**EventHelix.com/EventStudio 1.0**
client		server 1	router	server 2		
RAM software	ROM software	BOOTP Server 1 / TFTP Server 1	BOOTP Relay Agent	BOOTP Server 2	TFTP Server 2	10-Feb-02 23:08 (Page 4)

BOOTP_Reply ← hw_address, client_ip, UDP_Port = N68, boot_file_name	The BOOTP Relay agent then uses the hardware address to directly send the BOOTP reply to the client. This message is sent to UPD port 68
Store self IP address	Client stores its own IP address. This IP address will be used in all communications with the outside world
Enable ARP	Since the client has received its IP address, it can participate in ARP
begin Boot file download using TFTP	Now the client can initiate boot file loading. Typically TFTP is used in downloading the file. TFTP is selected over FTP as it uses UDP. This minimizes the amount of protocol code that needs to be burst into the client ROM.
TFTP → file = boot_file_name	Boot file name specified in BOOTP Reply is used to initiate TFTP
TFTP_Data ←	Complete contents of the file are loaded
TFTP_Data ←	
TFTP_Data ←	
TFTP_Data ←	
TFTP_Data ←	
end Boot file download using TFTP	TFTP session has ended. Now control will be transferred to the software that has just been downloaded.
Unzip and load downloaded file in RAM	The downloaded file is uncompressed and loaded into RAM
create ←	ROM software transfers control to the RAM software. The downloaded software includes the OS as well as the application, so no other downloads are required
	Note that the ROM software will include a complete implementation of the TCP/IP stack that would replace the primitive stack in the ROM.
Load and Initiate the OS	The Downloaded OS is booted
Start Application	The Downloaded application is started