Network Working Group
Request for Comments: 906

Ross Finlayson
Stanford University

June 1984

Bootstrap Loading using TFTP

Status of this Memo

It is often convenient to be able to bootstrap a computer system from a communications network. This RFC proposes the use of the IP TFTP protocol for bootstrap loading in this case.

This RFC specifies a proposed protocol for the ARPA Internet community, and requests discussion and suggestions for improvements.

Introduction

Many computer systems, such as diskless workstations, are bootstrapped by loading one or more code files across a network. Unfortunately, the protocol used to load these initial files has not been standardized - numerous methods have been employed by different computer manufacturers. This can make it difficult, for example, for an installation to support several different kinds of systems on a local-area network. Each different booting mechanism that is used must be supported, for example by implementing a number of servers on one or more host machines. This is in spite of the fact that these heterogeneous systems may be able to communicate freely (all using the same protocol) once they have been booted.

We propose that TFTP (Trivial File Transfer Protocol) [6] be used as a standard protocol for bootstrap loading. This protocol is well-suited for our purpose, being based on the standard Internet Protocol (IP) [4]. It is easily implemented, both in the machines to be booted, and in bootstrap servers elsewhere on the net. (In addition, many popular operating systems already support TFTP servers.) The fact that TFTP is a rather slow protocol is not a serious concern, due to the fact that it need be used only for the primary bootstrap. A secondary bootstrap could use a faster protocol.

This RFC describes how system to be booted (called the "booter" below) would use TFTP to load a desired code file. It also describes an existing implementation (in ROM) for Ethernet.

Note that we are specifying only the network protocols that would be used by the booting system. We do not attempt to mandate the method by which a user actually boots a system (such as the format of a command typed at the console). In addition, our proposal does not presuppose the use of any particular data-link level network architecture (although the example that we describe below uses Ethernet).

Network Protocols used by the Booting System

To load a file, the booter sends a standard TFTP read request (RRQ) packet, containing the name of the file to be loaded. The file name should not assume any operating system dependent naming conventions (file names containing only alphanumeric characters should suffice). Thereafter, the system receives TFTP DATA packets, and sends TFTP ACK and/or ERROR packets, in accordance with the TFTP specification [6].

TFTP is implemented using the User Datagram Protocol (UDP) [5], which is in turn implemented using IP. Thus, the booter must be able to receive IP datagrams containing up to 524 octets (excluding the IP header), since TFTP DATA packets can be up to 516 octets long, and UDP headers are 8 octets long. The booting machine is not required to respond to incoming TFTP read or write requests.

We allow for the use of two additional protocols. These are ARP (Address Resolution Protocol) [3], and RARP (Reverse Address Resolution Protocol) [1]. The possible use of these protocols is described below. The booter could also use other protocols (such as for name lookup), but they should be IP-based, and an internet standard.

The IP datagram containing the initial TFTP RRQ (and all other IP datagrams sent by the booter) must of course contain both a source internet address and a destination internet address in its IP header. It is frequently the case, however, that the booter does not initially know its own internet address, but only a lower-level (e.g. Ethernet) address. The Reverse Address Resolution Protocol (RARP) [1] may be used by the booter to find its internet address (prior to sending the TFTP RRQ). RARP was motivated by Plummer's Address Resolution Protocol (ARP) [3]. Unlike ARP, which is used to find the 'hardware' address corresponding to a known higher-level protocol (e.g. internet) address, RARP is used to determine a higher-level protocol address, given a known hardware address. RARP uses the same packet format as ARP, and like ARP, can be used for a wide variety of data-link protocols.

ARP may also be used. If the destination internet address is known, then an ARP request containing this address may be broadcast, to find a corresponding hardware address to which to send the subsequent TFTP RRQ. It may not matter if this request should fail, because the RRQ can also be broadcast (at the data-link level). However, because such an ARP request packet also contains the sender's (that is, the booter's) internet and hardware addresses, this information is made available to the rest of the local subnet, and could be useful for routing, for instance.

If a single destination internet address is not known, then a special 'broadcast' internet address could be used as the destination address in the TFTP RRQ, so that it will be received by all 'local' internet hosts. (At this time, however, no standard for internet broadcasting has been officially adopted. [**])

An Example Implementation

The author has implemented TFTP booting as specified above. The resulting code resides in ROM. (This implementation is for a Motorola 68000 based workstation, booting over an Ethernet.) A user wishing to boot such a machine types a file name, and (optionally) the internet address of the workstation, and/or the internet address of a server machine from which the file is to be loaded. The bootstrap code proceeds as follows:

(1) The workstation's Ethernet address is found (by querying the Ethernet interface).

(2) If the internet address of the workstation was not given, then a RARP request is broadcast, in order to find it. If this request fails (that is, times out), then the bootstrap fails.

(3) If the internet address of a server host was given, then broadcast an ARP request to try to find a corresponding Ethernet address. If this fails, or if a server internet address was not given, then the Ethernet broadcast address is used.

(4) If the internet address of a server host was not given, then we use a special internet address that represents a broadcast on the "local subnet", as described in [2]. (This is not an internet standard.)

(5) A TFTP RRQ for the requested file is sent to the Ethernet address found in step (3). The source internet address is that found in step (2), and the destination internet address is that found in step (4).

Note that because several TFTP servers may, in general, reply to the RRQ, we do not abort if a TFTP ERROR packet is received, because this does not preclude the possibility of some other server replying later with the first data packet of the requested file. When the first valid TFTP DATA packet is received in response to the RRQ, the source internet and Ethernet addresses of this packet are used as the destination addresses in subsequent TFTP ACK packets. Should another server later respond with a DATA packet, an ERROR packet is sent back in response.

An implementation of TFTP booting can take up a lot of space if care is not taken. This can be a significant problem if the code is to fit in a limited amount of ROM. However, the implementation described above consists of less than 4K bytes of code (not counting the Ethernet device driver).

Acknowledgements

The ideas presented here are the result of discussions with several other people, in particular Jeff Mogul.

References

[1] Finlayson, R., Mann, T., Mogul, J. & Theimer, M., "A Reverse Address Resolution Protocol", RFC 903 Stanford University, June 1984.

[2] Mogul, J., "Internet Broadcasting", Proposed RFC, January 1984.

[3] Plummer, D., "An Ethernet Address Resolution Protocol", RFC 826, MIT-LCS, November 1982.

[4] Postel, J., ed., "Internet Protocol - DARPA Internet Program Protocol Specification", RFC 791, USC/Information Sciences Institute, September 1981.

[5] Postel, J., "User Datagram Protocol", RFC 768 USC/Information Sciences Institute, August 1980.

[6] Sollins, K., "The TFTP Protocol (Revision 2)", RFC 783, MIT/LCS, June 1981.

[**] Editor’s Note: While there is no standard for an Internet wide broadcast or multicast address, it is strongly recommended that the "all ones" local part of the Internet address be used to indicate a broadcast in a particular network. That is, in class A network 1 the broadcast address would be 1.255.255.255, in class B network 128.1 the broadcast address would be 128.1.255.255, and in class C network 192.1.1 the broadcast address would be 192.1.1.255.