Network Working Group
Request for Comments: 1159

R. Nelson
Clarkson University

June 1990

Message Send Protocol

Status of this Memo

This RFC suggests an Experimental Protocol for the Internet community. Hosts on the Internet that choose to implement a Message Send Protocol may experiment with this protocol. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Discussion

The Message Send Protocol is used to send a short message to a given user on a given terminal on a given host. This is similar to the service provided by Unix's write command, which is limited to the users on that host. This service is also known on some hosts as "SEND".

As the Internet grows, more and more people are using hosts that do not run TCP/IP at all times. These hosts may be able to use a simple protocol that can be implemented in a subset of TCP/IP. The Message Send Protocol is one such protocol.

Note that a message sending protocol is already defined using TCP. The SMTP protocol includes a "SEND" command that will direct mail to a user's terminal. SMTP's SEND is not useful in this instance because TCP requires quite a bit of code. For the purposes of standardization, we will include a TCP based Message Send Service.

TCP Based Message Send Service

One message send service is defined as a connection based application on TCP. A server listens for TCP connections on TCP port 18. Once a connection is established a short message is sent by the client out the connection (and any data received by the client is thrown away). The client closes the connection after sending the message.

UDP Based Message Send Service

Another message send service is defined as a datagram based application on UDP. A server listens for UDP datagrams on UDP port 18. When a datagram is received by the server, an answering datagram

is sent back to the client containing exactly the same data.

Message Syntax

The message should consist of several parts. The first part is a single octet indicating the protocol revision, currently decimal 65, 'A'. The second part is the name of the user that the message is directed to. This and the remaining parts are null-terminated, and consist of eight-bit characters. Do not strip the eighth bit of the characters. The third part is the name of the terminal. The fourth part is the actual message.

The total length of the message shall be less than 512 octets. This includes all four parts, and any terminating nulls.

If the terminal part is empty, then "the right" terminal is chosen. If the user part is empty, then the message is written on the console.

If this protocol is changed, the revision number will be changed. In no case will any of the four parts be removed.

Advisories

It is advisable for servers to strip escape sequences before sending them to actual terminals. Some terminals can do nasty things when you send them certain escape sequence.

In both the TCP and UDP versions of the service, checksums are always used.

Security Considerations

Security issues are not addressed in this memo.

Author's Address

Russell Nelson
Educational Computing System
Clarkson University
Potsdam, NY 13699-5730

Phone: (315) 268-6455

EMail: nelson@sun.soe.clarkson.edu