

TCP Implementation Status

Jon Postel

Mike Wingfield

Norm Abramovitz

Bob Braden

Gary Grossman

Bill Plummer

Jim Mathis

Aage Stensby

Dave Clark

3 May 1979

Information Sciences Institute  
University of Southern California  
4676 Admiralty Way  
Marina del Rey, California 90291

(213) 822-1511

TCP IMPLEMENTATION STATUS

Introduction

This is a collection of reports on the status of implementations of the Transmission Control Protocol (TCP) [1]. The systems and reporters are:

- BBN Unix -- Mike Wingfield
- Ford KSOS -- Norm Abramovitz
- UCLA 360 -- Bob Braden
- DTI Unix -- Gary Grossman
- BBN Tenex -- Bill Plummer
- Tops-20
- SRI LSI-11 -- Jim Mathis
- NDRE NORD-10 -- Aage Stensby
- MIT Multics -- Dave Clark

BBN Unix Report - Michael A. Wingfield  
-----

The following is a status report on the TCP developed at BBN for DCA under contract DCA100-78-C-0011.

Implementation Language

C

Memory requirements

Bytes: 22000 instr, 15000 data

Number of connections

10 currently, but a compile time parameter

Operating System

Unix with BBN IPC additions

Hardware

PDP-11

Status

TCP has been completed since March 15, 1979.

User Programs

There is a THP user and server available, as well as a Telnet user and server. A trivial ftp user and server for the raw movement of bytes between files exists and has been used to move files between a Unix on the ARPANET and a Unix on the RCCNET. There is also a test package for exercising TCP.

Bandwidth

12Kb - 13Kb looped through the IMP, 20Kb internally looped.

Programming Effort

6 man-months

3 May 1979  
IEN 98

TCP Implementation Status  
BBN Unix

#### Documentation

"TCP/PSIP Development Report" which describes the user interface to TCP and high level design. A software documentation report is required under the contract but is not available yet.

#### Accessibility

TCP is always available on link 155 at BBN-Unix (Host 0, Imp 63). A THP server is always listening. TCP and THP are also available at EDN-Unix when their machine is on the net (Host 3, Imp 20). All numbers are decimal.

#### Additional Features

This TCP implements the AUTODIN II security, precedence, and TCC features specified in the SRI TCP report. The gateway/2 program can forward incoming segments to other destinations if necessary. A flakiness simulator in the gateway can be made to drop, duplicate, reorder, and break different percentages of outgoing segments for testing purposes. These parameters can be changed dynamically.



TCP Implementation Status  
Ford KSOS

3 May 1979  
IEN 98

Ford KSOS Report - Norm Abramovitz  
-----

Implementation Language

C and Modula

Memory Requirements

??

Number of Simultaneous Connections

will depend on system capacity (not table space limited)

Operating System

KSOS Kernel

Hardware

PDP-11/70, 11/45(?)

Status

Currently in detail design, implementation to start circa 1 May 79.

Higher Level Protocols

none yet. (Server protocols for multi-level secure systems need some serious research work yet.)

Other Features

Design envisages support for a multi-level secure network, although no specific support is included. Internet datagram layer is supported including fragment reassembly.

Point of Contact

Jay McCauley or Norm Abramovitz (chief designer/KSOS TCP)

UCLA 360 Report - Bob Braden  
-----

Philosophical Remarks

This implementation of the Internet and TCP protocols is designed to meet the following general objectives:

- (a) operate within the existing NCP system job, sharing code and control-block formats wherever possible;
- (b) be compatible at the system-call level with the existing user-level protocol modules;
- (c) implement the Internet protocol as a distinct layer, with interfaces designed to expedite the implementation of other higher-level internet protocols in addition to TCP;
- (d) require minimum NCP resources when internet protocol is not in use.

Hardware

IBM 360/370, with a Santa Barbara interface to the IMP.

Operating System

OS/MVT, Release 21.8, with the addition of several user-written Supervisor-call routines (including the Exchange program). The UCLA NCP operates as a system job, with its own internal multiprogramming and resource management mechanism.

Implementation Language

BAL (IBM's macro assembly language)

Code Size (addition to existing NCP code)

|                          |                      |
|--------------------------|----------------------|
| Resident Control Process | 4K bytes             |
| Internet Protocol Layer  | 8K bytes (transient) |
| TCP Protocol Layer       | 9K bytes (transient) |

### Fixed Table Space

The limited fixed table space is included in the code (above).

### Connections Supported

Only practical limitation is amount of memory available in NCP region for buffers and per-connection control blocks (see following).

### Cost per Connection

#### Control blocks

For each connection, the internet and TCP layers require control blocks totaling 256 bytes.

#### (\* )Receive

Segment reassembly buffer = max segment size - min internet header length + 16 = 572 bytes per buffer.

#### (\* )Send

128 bytes per unacknowledged segment.

Note: The actual data being sent is not counted here, as it occupies buffer space belonging to the appropriate user-level protocol module.

(\* )Note: There is a dynamic pool of these objects, shared among all active connections. The pool grows and shrinks dynamically with the number of connections; it is probably reasonable to expect an average of one segment reassembly buffer and one unacknowledged segment (total of 700 bytes) per TCP connection.

In addition to this TCP-specific memory, there is the memory to support the user-level protocol. For example, a server-Telnet session to TSO requires control blocks and buffers totaling about 1800 bytes; this is identical for TCP and for the ARPANET Host-Host protocol.

### Performance

No information available yet

3 May 1979  
IEN 98

TCP Implementation Status  
UCLA 360

Operational Status

In debugging

Implementation Time

Approximately 20 man-months

User-Level Protocols Available

User and Server Telnet

Documentation

In progress

DTI Unix Report - Gary Grossman  
-----

Background

Digital Technology Incorporated (DTI) is constructing an IOC Network Front End (INFE) to connect WWMCCS hosts and terminals to the AUTODIN II network. As part of the INFE effort, DTI has developed a version 3 TCP which will be updated to a version 4 TCP when Government direction is forthcoming.

Implementation Language

PDP-11 Version 6 Unix C

Instruction Space Memory Required

9.6K 16-bit words of PDP-11 instructions

Buffer and Table Memory Space Required

Fixed

4K 16-bit words

Per-connection

Words of tables, etc.

100

Words of buffer space

None (uses user process buffers)

Connections Supported

Order of 50

Operating System

Version 6 Unix as modified for the INFE by the addition of a special interprocess communication mechanism.



3 May 1979  
IEN 98

TCP Implementation Status  
DTI Unix

#### Hardware

DEC PDP-11/70 and PDP-11/45 (Code not machine-dependent; could run on other machines given appropriate C compiler and operating system, but there may be address space problems on machines with a single 16-bit address space per process.)

#### Operational Status

Has been run extensively with identical implementations only.

#### Telnet Availability

None

#### Other Protocols Available

Terminal-to-Host Protocol (THP) as specified in

Postel, J., Garlick, L., and Rom, R., Terminal-to-Host Protocol Specification, Stanford Research Institute, Menlo Park, 15 July 1976.

#### Measured Bandwidth

Maximum bandwidth on a single connection through the DTI H516 IMP is 60K bits/second.

#### Measured Packet Processing Rate

Order of 40 packets/second

#### Implementation Time

Six (6) man-months

#### Documentation

Described in

Healy, D.C. and Buford, W.E., Transmission Control Protocol and Pseudo Segment Interface Protocol Program Maintenance Manual, DTI Document Number 78020.C-INFE.3, Digital Technology Incorporated, Champaign, Illinois, 18 December 1978, 207 pp.

There is also extensive documentation in the source.

DEC PDP-11/70 and PDP-11/45 (Code not machine-dependent; could run on other machines given appropriate C compiler and operating system, but there may be address space problems on machines with a single 16-bit address space per process.)

Operational Status

Has been run extensively with identical implementations only.

Teletype Availability

None

Other Protocols Available

Terminal-to-Host Protocol (THP) as specified in

Postel, J., Garlick, L., and Row, R., Terminal-to-Host Protocol Specification, Stanford Research Institute, Menlo Park, 15 July 1976.

Measured Bandwidth

Maximum bandwidth on a single connection through the DTI 8516 IMP is 60K bits/second.

Measured Packet Processing Rate

Order of 40 packets/second

Implementation Time

Six (6) man-months

Documentation

Described in

Healy, D.C. and Buford, W.E., Transmission Control Protocol and Pseudo Segment Interface Protocol Program Maintenance Manual, DTI Document Number 78020-C-INT-3, Digital Technology Incorporated, Champaign, Illinois, 18 December 1978, 207 pp.

BBN TENEX/TOPS20 Report - William W. Plummer  
-----

Background Information

There are several TCP implementations for TENEX and TOPS20. The first such version was written in (the compiler language) BCPL and ran in user mode. Later a MACRO-10 (assembly language) version was written which has been included in the monitors. The BCPL version tracked all protocol changes through TCP 2.5.2 and the assembly language version started with TCP 2.4. The BCPL TCP has now been retired and assembly language version are available for versions 2.5.2, 4.0 and 4.0(ver 1) (see below).

It should be noted that the higher level protocols such as Telnet were implemented originally for the BCPL TCP. Since only minor changes have been made to the user interface, it is these same programs which are still in use with the monitors TCP.

Bandwidth figures are rather hard to give because of variations in the operating systems, networks, setting of various parameters in the TCP during the measurement runs, and the continuing evolution of the TCP code itself. The bench mark used for the figures below is TCPTST, a program which sends 1000. packets to itself. This program attempts to keep 8 packets outstanding and thus something between 125. and 1000. acknowledge packets will flow in the reverse direction.

BCPL TCP 2.5.2

Code size

20,000. words

Data space

16,000. words

Environment

KA10 processor

TENEX operating system

JSYS traps used to simulate monitor calls SNDIM/RCVIM to communicate with ARPANET

Implementation time

Basic protocol module

4 man-months, coding 6 man-months, debugging

Each server

1 man-month, coding 1 man-month, debugging

Higher-level protocols implemented

Telnet server

Telnet user

ECHO

SINK

PRDATA (measurement program for PR Net)

Bandwidth bench mark

438 seconds

Status

Retired, but could easily be revived.

Comments

Debugging was relatively easy since it was done in user mode. On the other hand only on other host was available to talk to. Many debugging feature such as a packet tracer were incorporated which helped considerably. Being in BCPL, the TCP implementors occasionally found themselves at the mercy of the compiler implementors.

Assembly language TCP 2.5.2

Code size

5,000. words

Data space

4,000. words on TOPS20 101B 16,000. words on TENEX  
16,000. words on TOPS20 3A on 2020

3 May 1979  
IEN 98

TCP Implementation Status  
BBN TENEX/TOPS20

#### Environment

KA10 processor running TENEX or, Model A KL10/20 running TOPS20 101B  
or, Model 2020 running TOPS20 3A.

#### Implementation time

Basic protocol module

2 man-months, coding 6 man-months, debugging

Each server

None. (Same programs used)

Higher-level protocols implemented

Same as above.

#### Bandwidth bench marks

46 seconds on KA10 67 seconds on 2020 30 seconds on KL10 with cache

#### Status

Currently available on BBNC, ISIC, and SRI-KA. Secure version  
running on BBNE and ISIA.

#### Comments

Last version 2.5 implementation. Many intermediate version 2.X  
steps were taken over a relatively long time. Debugging hampered by  
lack of stand alone computer time for monitor debugging.

#### TCP 4.0

This is the first of the version 4 implementations. The code was  
derived from the version 2.5.2 sources in approximately 2 weeks.  
Bandwidth and size figures are the same.

#### Status

Running on BBNB



### TCP 4.0(ver 1)

TCP 4.0(ver 1) is derived from TCP 4. Internally it is somewhat different and offers more features to users. Most notable is a set of JSYS calls by which users may access the Internet layer in order to experiment with private Internet protocols. The mechanism by which TCP transfers data between user buffers and Internet packets is considerably different than it was in 4.0 and is much faster. Also the free storage management package has been streamlined. The major task underway is embedding the Telnet server in the monitor along with the TCP itself.

#### Code size

5,200 words

#### Data space

256,000 words on KL20 TOPS20 3A 16,000 words on TENEX, 2020, TOPS20 101B

#### Environment

##### Development System

KL2040 processor  
TOPS20 3A monitor  
(Should work with TENEX and other versions of TOPS20, but this has not been tried yet.)

##### Bandwidth bench mark

26 seconds (2040 without cache) 43 seconds (2020)

#### Status

Running on BBNF by arrangement

SRI LSI-11 Report - Jim Mathis

-----  
Implementation Language

PDP-11 MACRO assembler

Hardware

LSI-11 or PDP-11 in user mode

Note: The software was developed to run on the MOS operating system for LSI-11s, but should be transportable with minor changes to other real-time operating systems that provide an interprocess communication and signaling mechanism. An earlier version of the software (version 2.5) has been transported to RSX-11, ELF, and Unix.

Instruction Space Requirements

TCP

approximately 1800 decimal words

Internet Protocol

approximately 1000 decimal words

Note: The TCP and Internet protocol operate from a network characteristics table that simplifies conversion from use on one packet-switched network to another. The software currently supports attachment to the ARPANET or the Packet Radio Network. The software is structured to allow attachments to multiple networks, although this feature has not been tested.

Simultaneous Connections

limited only by CPU loading and the amount of buffer space available

The state information for each TCP connection is maintained in a connection block of approximately 50 words; but a minimum of 100-500 words of buffer space is recommended for each connection in active use.

Comment

The TCP and Internet software were developed for the Packet Radio Terminal Interface Unit to provide terminal access to internetwork hosts (a function similar to that provided by the TIPS for ARPANET hosts). Because the TIU is not a timesharing machine, the only higher level protocols available are Telnet. The Telnet will operate, by manual command, in either the active (user) or passive (server) mode.

Status

TCP

- in preliminary release stage and undergoing further debugging
- has undergone functional tests with other TCP version 4 sites

Internet

- has undergone functional tests with other TCP version 4 sites

Implementation Time

The effort expended to design and code the TCP and Internet software is difficult to estimate; the software has been evolving for the last 3 years. Reconstruction of the software from scratch would probably require about 6 to 10 man-months.

3 May 1979  
IEN 98

TCP Implementation Status  
NDRE NORD-10

NDRE NORD-10 Report - Aage Stensby  
-----

## TCP STATUS

### Philosophical Remarks

Due to the idiosyncrasies of the operating system, it has been a goal to keep the number of processes as low as possible without complicating the internal logic of the program. Accordingly, there is one process for input handling and one process for output handling. In addition, it has been necessary to include a separate retransmission process. The code has been tested rather thoroughly routine by routine and connections have been opened and closed a number of times in loop mode. The TCP does not yet support urgent or rubber EOL.

### Hardware

The TCP runs on a NORD-10 with 64K memory.

### Operating System

SINTRAN III, version 77.05.17c

### Implementation Language

NORD PL, which is a machine oriented medium level language.

### Code Size

7k words(16 bit) This includes the present internet module.

### Buffer Space

Dynamically allocated, maximum 7k which may be extended.

### Connections Supported

No strict limit, but probably in the order of 4-5 due to available processing power.

### Cost per Connection

?

Delay per Packet

Not measured

Bandwidth

Not measured

CPU Utilization

Not estimated

INTERNET STATUS

The internet code is not separated from the TCP code. It is planned to gather it in a separate process with a well-defined interface to the TCP. Reassembling of fragments is not yet done. Maximum acceptable segment size is 1008 bytes.

The internet address is:

| IMP | LH | HOST |
|-----|----|------|
| 51  | 0  | 1    |



3 May 1979  
IEN 98

TCP Implementation Status  
MIT Multics

MIT Multics Report - Dave Clark  
-----

Implementation Language

PL/I

Memory Space (in 36 bit words)

TCP 13.6k, IN 5.5k

Table Space

About 40 words per connection

Buffer Space

In virtual memory, no relevant limit in size

Hardware/Operating System

H68/80, Multics

Operational Status

Currently available about half time for experimentation, up as experimental service in week or so.

Other Protocols

Datagram protocol, name server, user Telnet, server Telnet (The server Telnet is a kludge and requires programming by Honeywell before it can be considered really useable. Performance: No relevant data available yet. In a subjective test, Telnet is useable, but sometimes sluggish.

Implementation Time

I have no idea. It has been worked on by several people at the 10% level, so the total time invested is quite unclear.

References

- [1] J. Postel. "Transmission Control Protocol -- Version 4," IEN 81, USC-Information Sciences Institute, February 1979.