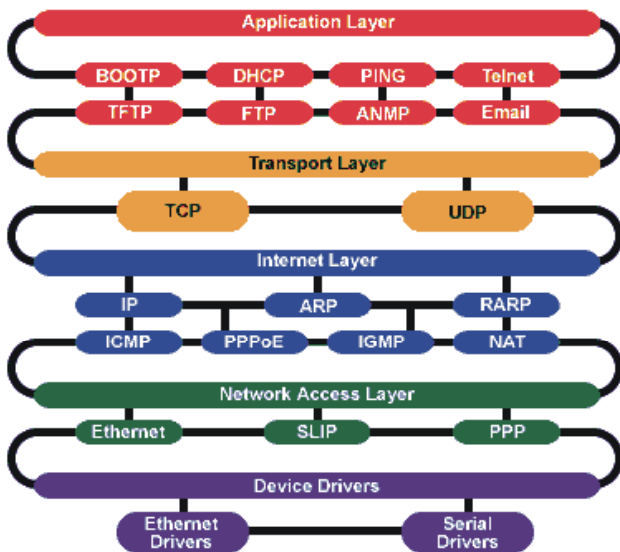




USNet is a processor and RTOS independent TCP/IP networking stack designed for use with real-time, embedded applications. It includes example clients and servers. It provides drop-in support for many processors and tool chains. Complete source code and test programs are provided. USNet is licensed per application and is royalty free.

Components

USNet comes with complete source code that covers all layers in a TCP/IP network stack, ranging from example applications to device drivers.



Application Layers

- BOOTP* Boot client. Allows client to obtain its IP address and boot file.
- DHCP* Dynamic Host Control Protocol. Allows a client to obtain a temporary IP address. c/s.
- PING* Uses ICMP to report if a host is responding. c.
- Telnet* Telnet client acts as a terminal emulator. Telnet server acts as a command processor.

FEATURES

- Embedded TCP/IP
- Works with RTOS or standalone. See list of supported RTOSes below.
- Processor independent
- User Configurable
- Royalty Free
- Small footprint (24K - 39K bytes code space depending on the processor)
- Many drivers supplied including Ethernet and serial drivers
- ROMable and reentrant
- Includes complete source code in ANSI C

- TFTP* Trivial File Transfer Protocol. c/s.
- FTP* File Transfer Protocol. c/s.
- SNMP Simple Network Management Protocol agent v1, v2, and v3.
- Email Email send (SMTP) client, receive (POP3) client, and MIME to attach data files.

Transport Layer

- TCP* Transmission Control Protocol – for reliable connection. Required for Telnet, FTP.
- UDP* User Datagram Protocol – unreliable, connectionless.

Internet Layer

- IP* Internet Protocol
- ARP* Address Resolution Protocol – IP to Ethernet address. c/s.
- RARP* Reverse Resolution Protocol – Ethernet to IP address. c/s.
- ICMP* Internet Connection Management Protocol
- IGMP* Internet Group Management Protocol (multicast)

*Included in basic USNet package.
c = client, s = server

PPPoE PPP over Ethernet. c/s.
NAT Network Address Translation – allows sharing an IP address. c/s.

Network Access (Link) Layer

Ethernet* Interface to Ethernet drivers. Process Ethernet protocol.
SLIP* Serial Line Internet Protocol
PPP Point to Point. Includes CHAP and MS-CHAP authentication protocols.

Device Drivers

Ethernet Popular Ethernet controllers are supported.
Serial UART drivers are provided.

Advantages

Flexible Configuration

Because code and data space are at a premium in an embedded system, USNet can be configured to use only those clients, servers, protocols, device drivers and link layers needed by your application. By judiciously selecting features and capabilities, the USNet TCP/IP stack can be reduced to as little as 24K code space, depending on the processor.

Clean Design

USNet implements a clean, modular design. Network layers provide a standardized API that presents a common set of entry points for each layer in the network from the Transport layer down to the device driver.

Choice of API

Network applications can be written to use either the well known Sockets API, or the USNet Dynamic Protocol Interface. The DPI is a simplified API that takes little code space and is easy to use. Sample applications demonstrating both of these APIs are included in USNet.

Zero-Copy

USNet supports a zero-copy interface, for efficiency. Information to be sent is assembled once and sent directly by the network controller. Likewise, received

data is stored by the network controller, and then read from the same memory by the application. The network controller must support DMA. Use of the zero-copy interface is demonstrated in example code that ships with USNet.

Routing

USNet supports up to 256 network interfaces, and will route packets between interfaces.

Built in Tests

USNet comes with a full set of tests and debugging aids that allow you to be confident of proper operation on your platform. Several state display functions allow more easily checking the operation of features implemented in your application and tracking down unexpected behavior.

Ethernet Drivers

USNet provides your choice of device drivers and link layers. Device drivers provide support for most popular network controllers including fast (100 MB) Ethernet devices.

- 3COM 3C5xx, 3C9xx
- AMD AM186ES, AM7996x
- Crystal CS8900
- DEC DC21x40
- Freescale MPC860
- Intel i8255x, i8259x
- Novell NE2000
- SMSC LAN91C111

Add-ons

USNet® add-ons enhance real-time, embedded TCP/IP protocol stack with Internet and Web enabling technology. This capability may be used on private LAN/WAN networks or on the Internet.

Email

USNet Email includes support for SMTP, POP, and MIME. These protocols enable any embedded device to send and receive email.

NAT

USNet NAT, which is written based on industry standard RFC 2663, enables a local-area network (LAN) to use one set of IP addresses for internal traffic and a second set for external traffic.

- It allows various types of devices to share a single IP address and combine multiple connections into a single Internet connection.
- It also allows new devices to be added to the network with no modifications or configuration to a typical NAT-enable router.
- It provides a type of firewall by hiding internal IP addresses on a private network.

PPP

smxNS PPP is based on industry standard RFC 1661 for establishing a link to a single remote host, often via a serial or modem link. A dialer is included.

PPPOE

USNet PPPoE is based on industry standard RFC 2516 for connecting the users on an Ethernet network to the Internet through a shared access to a broadband medium, such as a single DSL line, wireless device or cable modem. USNet PPPoE can be configured to support operation as either a PPPoE Host or a PPPoE Access Concentrator. This means that USNet can be incorporated into a device that accesses the Internet through a PPPoE router, or it can be incorporated into a router, acting as an Access Concentrator that serves PPPoE hosts.

The USNet PPPoE implementation requires virtually no more knowledge on the part of the end user other than that required for standard Dialup Internet access. In addition, PPPoE requires no major changes in the operational model for Internet Server Providers (ISPs) and carriers.

SNMP Agent

Includes version 1, 2, and 3 support and a MIB compiler. The agent may be used with both MIB-II and private, user-created MIBs.

Web Server

The USNet Web Server allows an embedded system to present real-time information to desktop Web browsers. It is a full-feature HTTP server optimized

to run in 16/32 bit environments. It is processor, RTOS, and TCP/IP stack independent. A file system, operating system, and dynamic memory are not required. The USNet Web Server is HTTP 1.0 compliant and includes HTTP 1.1 features such as authentication and server-side includes.

The USNet Web Server makes an application dynamic with META commands and server-side includes for dynamic page content and the ability to add new web pages while the application is running.

Web pages can be created using standard web design tools and converted to a C file to link into the application, using an included utility.

Supported RTOSes

- SuperTask!/MultiTask!, TronTask!
- Hitachi HI-SH7
- Keil RTX166
- Motorola PPSM
- NEC RX850
- Threadx

USNet has a well-defined porting layer and can be ported to another RTOS or run standalone. The RTOS only needs the ability to: launch a task, signal an event, wait on an event with a timeout, disable and enable task preemption, and a timer with at least 50ms resolution.

Supported Processors and Compilers

<u>Processor</u>	<u>Compilers</u>
680x0/683xx	Crosscode, Diab, Microtec
80x86 RM	Borland, Microsoft, CAD-UL
80x86 PM	Borland, CAD-UL, MetaWare, Microsoft, Watcom
80960	GNU, Intel
ARM7	ARM
C16x	Keil
ColdFire	Diab
DragonBall	CodeWarrior, Crosscode
MIPS	EPI, GNU
PowerPC	Diab, Green Hills, MetaWare
SH1, 2	GNU
SH2, 3, 4	Hitachi
V850	Green Hills

Size and Performance

Configuration

<u>Setting</u>	<u>Name</u>	<u>Value</u>
Network Interfaces	NNETS	1
Connections	NCONNS	4
Route Table Entries	NCONFIGS	8
Frame Buffers	NBUFFS	5

Memory Requirements

<u>Component</u>	<u>Coldfire</u>		<u>ARM Thumb</u>		<u>X86 PM</u>	
	<u>ROM</u>	<u>RAM</u>	<u>ROM</u>	<u>RAM</u>	<u>ROM</u>	<u>RAM</u>
USNet Core Library	39K	11K	24.0K	10.2K	31.9K	9.8K
Socket API	+10.5K	+0.2K	+4.4K	+0.3K	+6.3K	+0.4K
DHCP	+4.8K	+0.4K	+3.3K	+0.1K	+5.3K	+0.1K
Telnet Server	+1.4K	+2.5K	+0.7K	+0.2K	+1.4K	+0.0K
NAT	+5.6K	+0.4K	+4.2K	+0.3K	+4.9K	+0.3K

These memory Requirements are typical for a system that services one active TCP session.

Notes

- The Core Library includes support for TCP, UDP, IP, ICMP, and an Ethernet driver.
- Socket API and DHCP support are commonly used but broken out separately.
- Support for IP Fragmentation and Reassembly is included.
- Support for IP Options Headers is not included.
- Support for Multicasting is not included.
- For each additional active session, USNet should be configured with NCONN increased by 1, NCONFIG by 1 and NBUFFS by 5. So each active session (client or server) adds about 8K RAM to the memory requirements. In order to support an additional connection, increase NCONN by 1, NCONFIG by 1 and NBUFFS by 5. Therefore, each additional add about 8K RAM to the requirements.

Network Performance

<u>Protocol</u>	<u>Test</u>	<u>App write size</u>	<u>PPro 150</u> <u>Rate, MB/s</u>	<u>K6-2 400</u> <u>Rate, MB/s</u>
DPI, TCP		512	5.4	7.2
DPI, TCP		1024	7.8	9.5
DPI, TCP		1460	9.1	10.7
Sockets, TCP		1460	9.3	10.5

Systems used in this Test

PPro 150

- Pentium Pro 150MHz
- Intel i82559 Ethernet controller

K6-2 400

- AMD K6-2 400MHz
- Intel i82559 Ethernet controller