Linux ARCNET Driver Information

ARCNET Card and Encapsulation Support

The Linux ARCNET driver has been part of standard Linux distributions since the release of the Linux 1.2 kernel in early 1995. Since then, the driver has evolved to support more types of ARCNET hardware chipsets and software protocols and encapsulations.

Linux Kernel	***ARCNET Driver***	***Supported Contemporary Controls ARCNET Cards***					***Supported Encapsulations***			
		PCX	***PCA 66***	***PCX 20***	***PCI 20***	***PCM 20***	***RFC 1201***	***RFC 1051***	***Ether***	***Raw***
1.2.x	***v1.11***	X	X				X			
	v2.22 (+)	X	X				X	X	X	
2.0.x	***v2.56***	X	X				X	X	X	
2.2.x	***v3.02***	X	X	X			X	X	X	
	v3.9x (+)	X	X	X	X	X	X	X	(*)	X
2.3.x/2.4.x	***v4.00 (!)***	X	X	X	X	X	X	X	(*)	X

Notes: (*) *This feature was temporarily removed in 3.9x beta versions, but will be restored by the time 4.00 is released.*
(+) *This driver version is not part of the standard Linux distribution and must be downloaded separately.*
(!) *Linux 2.3.x is a pre-release Linux version and is not recommended for general use. Linux 2.4.x was not yet released at the time this document was written.*

Table 1. *Features supported by different Linux driver versions.*

To use an ARCNET card under Linux, the user needs to know three things: the hardware chipset on the card, the encapsulation type(s) in use on the local network, and the transport protocol(s) to use (such as TCP/IP or IPX). The exact set of required features will determine which Linux kernel version is required, and whether an additional driver upgrade is required for that kernel.

• *Hardware Chipset Support*

Under a contract from Contemporary Controls, the most recent Linux ARCNET driver (3.9x/4.00) was extended to support all current Contemporary Controls ARCNET cards, including the following:

PCX	8-bit ISA	COM90C65
PCA66	16-bit ISA	COM90C66
PCX20	8-bit ISA	COM20020
PCI20/22	8-bit PCI	COM20020/22
PCM20/22	PCMCIA	COM20020/22

The PCX and PCA66 cards can be used with both memory-mapped and I/O-mapped buffers.

Version 3.9x of the driver is not currently part of any released Linux distribution, but it can be downloaded separately if the additional hardware support is needed. It will be part of Linux 2.4.0 once that kernel has stabilized.

The older ARCNET driver, included as standard in Linux 2.2-based distributions (such as Red Hat 6.1) is version 3.02. This version supports the Contemporary Controls PCX, PCA66, and PCX20 cards, but does not include support for plug-and-play operation and so is not compatible with the PCI20 (PCI) and PCM20 (PCMCIA) cards. It requires memory-mapped buffers except for the PCX20 card, and does not use the enhanced 16-bit bus transfers supported by the PCA66.

Older, obsolete Linux kernels numbered less than 2.2.0 also have verying degrees of ARCNET support, as described in the table above.

• *Encapsulation Types*

One aspect of ARCNET that is unfamiliar to most users of Ethernet networks is the requirement to choose an *Encapsulation Type*. The encapsulation type is implemented at a software layer between the physical hardware and the transport protocol (such as TCP/IP or IPX).

In an ideal world, there would only be one encapsulation type, and all transport protocols would work with that. However, for various reasons different software vendors have used different methods of storing protocol information in an ARCNET packet, and these methods are not compatible. For example, TCP/IP can be transported over ARCNET using any of RFC1201, RFC1051, or ethernet-style encapsulation, and released products have used all three types. The Linux driver is unique in that it supports all three of them, and an additional "raw" encapsulation that can be used to communicate with very low-level embedded systems.

RFC1201 - This is the most recently defined packet encapsulation type. It is also the most complex and offers the highest performance. Its name comes from a standards document, Request for Comments #1201, released by the Internet Engineering Task Force (IETF). Nearly all new ARCNET software systems use RFC1201-style packet encapsulation. The Linux driver can use RFC1201 to communicate using both TCP/IP and IPX. Novell Netware's ARCNET driver uses IPX over RFC1201.

RFC1051 - An earlier IETF standard, RFC1051 is a minimalistic encapsulation that includes only a protocol ID field at the beginning of the packet. The only defined protocol ID number in RFC1051 is for TCP/IP.

Ethernet-Encap - This encapsulation type was created by Microsoft for use with their NDIS2 and early NDIS3 ARCNET drivers (which were available for LAN Manager, Windows for Workgroups, and Windows 95). Ethernet-Encap consists of a special protocol ID field, followed by an Ethernet packet header, followed by the transport protocol data. The advantage of this encapsulation is it is easy to implement on top of an inflexible protocol stack (such as NDIS2). One major disadvantage is that 14 bytes of each packet are wasted by duplicated header information.

Raw - Packets sent with Raw encapsulation actually do not receive any encapsulation at all. It is up to the sender and receiver of the packet to arrange that packet types and protocols can be identified at each end. The raw encapsulation driver, available starting with the Linux ARCNET driver v3.9x, can be used by systems developers to allow Linux to communicate with existing non-standard ARCNET devices.

In Linux, it is possible to load any or all of the encapsulation drivers simultaneously. Beginning with the 3.9x driver, Linux will auto-detect the required encapsulation type to use for each host on the local network. In earlier driver versions, a separate network device (arc0, arc0s, arc0e for RFC1201, RFC1051, and Ethernet-Encap respectively) would be created for each encapsulation driver.

• *Transport Protocols*

Linux supports TCP/IP and IPX over ARCNET, as well as a "raw" access mode (using BSD sockets) that allows users to implement their own protocol types. Additionally, IPv6 support was added to the driver by a contributor, but his success has not been confirmed by anyone else.