The journal of Apple technology.

Volume Number: 15 (1999)
Issue Number: 10
Column Tag: Hardware

The Well-Connected Mac

by Rich Morin and Doug McNutt

Protocols, Plugs, Wires, and Waves

Background

Early Macintosh computers had very limited capabilities for communication. Over time, however, a number of interfaces have been added to the Mac's repertoire. Current Macs can communicate via ADB, AirPort, Audio, Ethernet (both 10 and 100Base-T), FireWire, IrDA, LocalTalk, PC Card, PCI, SCSI, USB, and Video interfaces, to name just the most common.

Obviously, the characteristics of these interfaces vary wildly. Most, but not all, are digital. Most are carried by "serial" cables, but some use parallel wiring, plug directly into the motherboard as cards, or communicate via wireless means such as infrared or radio waves. It is not surprising, therefore, that many Macintosh users (and even developers!) find themselves a bit confused at times.

This article addresses this confusion by giving brief (and somewhat subjective) descriptions of each interface, along with references for further inquiry. It won't explain how to program the Mac for any of these interfaces, let alone how to design your own interface hardware, but it should let you know a bit about the strengths (and weaknesses) of each interface.

Before getting into the descriptions themselves, however, we need to cover some basic terms that will show up later in the article. Feel free to skip over this material, but be sure to come back to it if you start to get confused. The interfaces have no compelling "family tree", and we will present them in roughly historical order.

The first distinction of interest is that of "analog" vs. "digital". All electronic interfaces are analog, in the sense that they use measurable voltages, currents, and other analog phenomena to communicate. Digital interfaces are distinguished, however, by the fact that their signals are interpreted solely as (sets of) bits. Thus, most audio and video interfaces are analog, but disk drives and such are always connected via digital interfaces.

The next distinction has to do with topology. Busses connect two or more devices together; ADB, SCSI, and PCI are all busses by this definition. "Point to point" interfaces, in contrast, tie pairs of devices together; most audio and video interfaces are used in this manner. Distinctions of this sort can become fuzzy, however, when "hubs" are introduced. For instance, 10Base-T is an implementation of a bus (Ethernet), but it is based on a "hub and spoke" topology.

Most interfaces use wires (in some form) to carry their signals. AirPort and IrDA, however, use radio and infrared waves. Even in the wired domain, there are important distinctions to be made. "Serial" interfaces (e.g., ADB, FireWire, and LocalTalk) send all of the data bits down a single wire; "parallel" interfaces (e.g., PCI and SCSI) use multiple wires, sending many bits at the same time.

Got all that? Great! Now, let's look at some interfaces...

Serial Ports

Macintosh serial ports get used for a variety of purposes: modems, printers, and LocalTalk connections. Consequently, they are a bit different from the serial ports found on PCs and other computers. Nonetheless, they interoperate quite well with most standard serial devices, as long as the cabling is done properly.

Serial ports on the Macintosh have always conformed to Radio Standard 422 (RS-422), which is a balanced-pair version of RS-232. Apple went a bit further, however, arranging its ports so that they could be connected to RS-232 devices using only wiring changes. Thus, the voltage applied to Apple's RS-422 pairs ranges from +5 to -5 volts. This is a bit high, but it is still within the RS-422 specification. It also allows an RS-232 device to be connected between the negative transmitter line (TxD-) and ground. The negative receiver line (RxD-) in the Apple port can be connected to an RS-232 transmitter if the positive receiver line (RxD+) is grounded.

Also provided are two housekeeping channels, one in and one out. They are driven (single-ended, in RS-232 fashion) from -5 to +5 volts. The RS-232 specification calls for two such channels each way. Consequently, there is a continuing question about whether to connect the wires to DTR or CTS, etc. The solution will forever depend on the connected device (e.g., modem or printer) and its associated support software. One more wire is available for external synchronization of the port, though it is rarely used for that. It has been used for carrier detect with modems when a Mac is used to support dial-in services.

The Mac128k provided power for external devices (at +5 and +12 volts) on its 9-pin connector. That feature was dropped with the Mac+ and all further Macs in order to use the mini-DIN 8 pin connector. Somewhere into the first PowerMacs, a special mini-DIN 9 connector was made available and power was restored for peripheral devices. The mini-DIN 9 socket will accept a mini-DIN 8 connector if power is not required.

The Zilog 8530 serial communications controller was used on early Macs and the technical manual for that chip remains the Apple standard. The 8530 has a feature which provides for mixing the clock with the serial NRZ data stream to create what Zilog calls FM0 modulation. The effect is to move the frequency band occupied by the data away from zero frequency, which requires a direct connection, up to 128 kHz, which can pass through transformers.

It is that modulation which makes AppleTalk connections possible between all Macs made prior to the iFruit. All one had to do was to connect the receive and transmit ports of a Mac to each other and, through a transformer, to the receive and transmit ports of another Mac. In fact, you could connect up to 32 transformers between Macs, using a two wire bus which needed only to be terminated at each end with 110 ohm resistors.

It turns out that transformers are not always necessary for AppleTalk connections. If the devices to be connected all share a common power source and are not subject to large ground fluctuations you can connect two Macs using an Apple serial cable between printer ports. It's also possible to make an AppleTalk bus without transformers.

Early Macs had to support video framing as a background task. Because the buffer for the printer port was guaranteed to be updated during video retrace periods, that port was chosen for use by AppleTalk software. In later machines, the modem port was also opened up for use.

The serial-port version of AppleTalk, now known as LocalTalk, has now been largely supplanted by the Ethernet-based version, EtherTalk. Running at something like fifty times the speed, EtherTalk is definitely worth considering for any network that can use it. By looking around a bit, you should be able to find economical hubs, adaptors, etc. DataCom Warehouse, in particular, carries a wide range of Ethernet hardware.

SCSI

The specification for the small computer system interface, SCSI, left a lot of room for variation in connectors; sadly, hardware vendors have taken far too much advantage of this ambiguity. Many of us have large boxes of SCSI cables, sporting assorted connectors: Blue Ribbon, DB-25, DB-50, etc. Apple has added at least two connector types (DB-25 and HDI-30) to this madness.

Apple's additions are particularly egregious because, in an apparent effort to save space, they eliminate the separate ground paths specified by the SCSI standard. If you have trouble getting Apple SCSI to work reliably, this "simplification" may be part of the problem.With ingenuity and a bit of luck, however, some combination of cables will allow you to add an arbitrary device to your SCSI "chain". If the termination gods are feeling generous at the moment, all of the devices on the chain may actually function!

In any event, Apple has opted to move away from SCSI, so the question is becoming moot for many Apple users. EIDE is taking on the role of connecting disk drives; FireWire and USB are handling scanners, digital cameras, and other exotica. With luck, SCSI will soon be nothing but a fading, noxious memory for most of us.

Apple employs SCSI as a single-ended parallel port in which each signal conductor is terminated with two resistors: 220 ohms to +5 volts and 330 ohms to ground. Actually, these resistors are only used in "passive" terminators; "active" terminators, which can yield better reliability, emulate this setup using solid-state devices.

In any event, the termination results in a quiescent potential of 3 volts and an effective termination resistance of 132 ohms (which matches ribbon cable). Both ends of the chain should be terminated, but for short runs such as a single internal disk, terminating one end is sufficient. Data bits are sent by grounding a signal line with a transistor.

The 5-volt terminator power must be supplied by at least one of the devices on the chain. The termination power is commonly supplied to the cable through a diode, which prevents multiple power supplies from "arguing" about the proper voltage. Consequently, the power supply with the highest voltage will end up supplying terminator power for the entire chain.

Some devices have DIP switches or jumpers to control termination; others attempt to make the determination automatically at startup time. Unfortunately, they may they get it wrong if all devices aren't powered up together.

In addition to the eight data conductors, there are several others which are used for device selection, synchronization, and other commands including arbitration (which allows any SCSI device to become a master, but is not used in the Mac OS). The data lines themselves are used to address devices, resulting in a limit of eight devices (numbered 0 through 7). The Macintosh is always unit seven.

In asynchronous mode with acknowledge, the bus's clock speed is three Mhz. Because the data path uses eight bits in parallel, this translates to 24 megabits per second. In synchronous mode, this increases to five MHz (40 megabits per second). Although faster (e.g., SCSI-2) devices can be used on a slow bus, their speed will be limited to that of the host's SCSI interface.

The SCSI specification has been extended with the introduction of Wide SCSI, SCSI-2, Fast SCSI, and Ultra SCSI. Wide can transfer as many as four bytes at a time. Fast SCSI runs at ten MHz. Unfortunately, Apple doesn't support any of these enhancements.

Thus, you're on your own to figure out how to connect a 68-pin Wide SCSI device to the Apple SCSI port. Generally, your best plan is to avoid such devices. If you have an application which really needs high-speed SCSI, you should purchase an add-in SCSI card, along with matching cables and terminators.

ADB

Although the Apple Desktop Bus is being replaced by the Universal Serial Bus (USB), there are still many millions of machines that use ADB, as well as a number of nice peripherals (e.g., Kensington's great Thinking Mouse) that are not (yet) available in USB versions. In short, ADB is not going to disappear overnight.

ADB has served Apple well. At ten kHz, it is plenty fast for keyboards, mice, and trackballs. The four-pin shielded mini-DIN connector seemed like a real step down from the 9-pin latched D connector used by the Mac128k, but it has proven sufficient for hot plugging and other insults.

The ADB was the first desktop interconnection scheme to support a microprocessor in the peripheral devices. It was thus able to count pulses in the mouse and transmit only the resulting number to the host (considerably reducing the needed bandwidth).

The main problem with ADB is not technical. Instead, it lies in the fact that ADB is not an industry standard. By switching to USB, Apple gains access to large numbers of devices (at least, prospectively) at mass-market prices. This was never going to happen with ADB, which has always been seen as an Apple-only standard.

Making matters worse, ADB even requires a license from Apple in order for vendors to use it. The license is inexpensive and is apparently intended only to let Apple control assignment of device types, but it may have been a deterrent. In any case, ADB never became an industry standard. USB, in contrast, is already built into many millions of PCs; now all the PC vendors need to do is tell the users what it's good for and how to access it.

ADB uses only three wires (ground, power, and data), though a fourth (for starting up the computer) shares the cable. The data line is pulled up to five volts by the host computer through a resistor, allowing any device to transmit data by simply grounding the wire.

Encoding is accomplished by dividing each bit period into thirds. Every bit begins with a downward transition. A rise to five volts at 1/3 of the bit time represents a zero and a rise at the 2/3 time represents a one. A message consists of a start bit, two to eight data bytes, and a stop bit. The computer supplies power at five volts (limited to 500 mA); this must be shared among all the connected devices.

ADB devices have addresses and registers which can be read and written by the host computer. Up to 16 devices are assigned addresses at startup time, using device types assigned by Apple. It is quite possible to have multiple devices (e.g., a mouse and a trackball) that use the same type and address.

Each ADB transaction begins with a command from the host, addressed to a device. There are four possibilities (talk, listen, reset, and flush) for each of the device registers. The host computer continually polls the "current" device (the one it has most recently talked with). Another device may request service by grounding the data line during the final stop bit of the message sequence. When the host computer detects a service request, it polls all connected ADB devices.

Audio

The Power Mac comes with stereo audio connectors for input and output. The internal representation of this data is CD quality (16 bits at 44,100 samples per second, supporting audio signals of up to 22 kHz). Although many audiophile purists say terrible things about CD-quality audio, most consumers find it to be quite adequate. In point of fact, so do we.

Unfortunately, Apple does not provide any convenient digital audio paths. It is possible to attach digital audio devices (e.g., CD, MP3, or mini-Disc recorders) via SCSI or FireWire, and some Apple machines provide an internal digital audio/video interface, but consumer devices cannot attach to any of this directly. This is disappointing; look at Sony's VAIO series to see a dramatically higher level of audio/video connectivity and support.

Motherboard Busses

Early expandable Macintosh systems used Texas Instrument's 10 MHz NuBus, a solid and well-designed bus that, unfortunately, never gained acceptance in the PC arena. Consequently, while Apple users simply plugged cards into available slots in their boxes, PC users had to fiddle with jumper settings, DMA vectors, and IRQ assignments.

More recently, however, Apple has adopted the Peripheral Component Interconnect (PCI) bus, which seems likely to be the standard for both PCs and workstations for some time to come. Be a bit careful, however, when you refer to the PCI bus. There are two available bus speeds (33, 66 MHz) and two available data paths (32, 64 bit).

Apple also uses the PC Card (aka PCMCIA) interface for its portable computers. Aside from noting that there are several types of PC Cards, in assorted thicknesses, we have no real reservations with the standard. It seems to be functional, overwhelmingly popular, and likely to be around for at least the next several years.

Video

Apple's early machines had video modulators which produced signals for American TV sets. This format - NTSC (National Television Standards Committee) video - has limitations which make it very unsuitable for use in high-resolution computer displays.

NTSC's horizontal resolution is limited to about 350 "lines" (700 pixels); the vertical resolution is theoretically capable of around 500 pixels, but many older sets only display about half that resolution. Even and odd horizontal rows of pixels are alternately displayed (interlaced), often producing an annoying jitter. Finally, some repetitive patterns (e.g., stripes) conflict with the phase encoding used for color information, yielding obnoxious artifacts.

Video monitors built especially for computers are not interlaced. They also have much higher bandwidth, and hence pixel count, than TV sets. As they have improved, however, a plethora of horizontal and vertical scan rates have appeared in the marketplace. Meaningful standards were never developed and compatibility nearly became impossible.

Apple attempted to deal with this problem by grounding selected leads in the video cable. Three pins were dedicated to this, supporting seven possible monitor types and a "no monitor connected" condition. Seven monitor types were not enough, however, so diodes were added to supply more detectable states. See Apple Developer Note HW 30 for an engaging account of this rework.

In the Wintel world, "multisync" monitors appeared to deal with this problem. In principle, these monitors sense the synchronization signals emitted from the computer and adapt appropriately. In practice, however, not all combinations work. In any case, these monitors do not supply the cable encodings desired by Apple computers. We have seen both partial and universal adapters which allows switching among some or all possible settings. Although these can be useful, they are not guaranteed to work in all situations.

Apple's latest monitor, the Apple Cinema Display (ACD), seems to be on a better track. Until the ACD, all flat panel monitors on the market were configured for connection to the computer's video output port. Consequently, the image was converted from digital to analog in the computer and then back to digital form in the monitor. This adds cost at both ends and reduces the accuracy of the displayed image.

The ACD does none of this; instead, it connects the display screen digitally to the computer. The result is that the update rate can be vastly reduced - to ten Hz or less - if the picture is static. If a pixel doesn't change, after all, there's no need to update it! Also, because digital signals are being sent over the cable, reflections and other artifacts of analog transmission can be eliminated.

It remains to be seen, however, whether other vendors will accept Apple's interface standard. At some level, it doesn't matter; Apple's interface card is PCI-based, so it should work on most desktop PCs. As other busses appear, new interface cards can be developed. On the other hand, some display vendors might choose to go directly to FireWire, making the interface card unnecessary and allowing a far larger number of monitors.

Of course, if every pixel on the display screen needs to change at once, the instantaneous bandwidth can become enormous. The ACD's 1.6 million pixels, at 24 bits each, comprise some 40 megabits of data. A FireWire connection, at 400 Mbps, could only handle ten such full-screen refreshes each second. Still, for many purposes, this would be more than adequate.

EIDE

EIDE (Enhanced Integrated Device Electronics) disk drives are inexpensive, often less than half the cost of similarly-sized SCSI drives. Part of this comes from simplicity; unlike SCSI drives, EIDE drives don't have built-in controllers. The computer must therefore tell the EIDE drive which cylinder, head, and sector to address, how to handle bad blocks, and more.

EIDE drives are also inexpensive, however, because of market considerations. Given current disk sizes (e.g., IBM's 36 GB DeskStar), one or two EIDE drives are likely to meet the needs of most desktop machines. So, EIDE has become the interface of choice for low-end service; SCSI is reserved for high-performance applications, servers, etc. Consequently, disk vendors no longer manufacture moderate-performance (and thus, economical) SCSI drives.

Unfortunately, EIDE's terrible distance (18") and fanout (2) limitations mean that it cannot be used for external devices. SCSI, the current standard for this role, isn't all that much better; the cables are awkward and very poorly standardized, the distance (20') and fanout (7) are limiting, and hot plugging is generally a risky business.

The EIDE bus is usually implemented with a 40-conductor ribbon wire and is limited to use inside the case of a computer. There is no provision for terminating the transmission path to eliminate reflections. Only very simple addressing (limited to two devices) is available and most of the disk management issues must be handled by the host computer.

Although USB is being proposed as an alternative to EIDE and SCSI, it is simply too slow to serve the needs of high-speed devices. We are therefore hoping that IEEE-1394 will answer the need for economical and flexible connectivity to block-oriented devices.

Ethernet

Two Ethernet cabling methods are currently in common use. 10Base-2 uses a single co-axial cable, daisy-chained between the systems on the network and terminated at both ends by a 50-ohm resistor. 10 and 100Base-T, in contrast, employ twisted-pair cabling between host machines and a central "hub".

Although both 10Base-2 and 10Base-T are specified as ten MHz networking technologies, this doesn't tell the whole story. If more than one Ethernet interface is trying to send data, the possible throughput goes down by about half (because of packet collisions, etc.) If higher-level protocols (e.g., NFS) react poorly to broken packets, the bandwidth can degrade even further.

Bandwidth also interacts somewhat with network topology. Because 10Base-2 uses a single coaxial cable for everything, only one packet can be "in transit" at a given time. Simple 10Base-T implementations (much like AirPort, described below) emulate this behavior, propagating the same limitations into the new (hub and spoke) cabling topology.

Fancier implementations (e.g., switching hubs) take advantage of the fact that a given packet only needs to travel along two of the spokes. Thus, A can send packets to B while C is sending packets to D. As the number of nodes increases, this optimization can improve performance dramatically. If you have a busy network and aren't using switching hubs, you aren't getting anything like full performance from your wiring!

When installing or upgrading a twisted-pair network, pay close attention to the type of cable you employ. Although 10Base-T works fine with Category-3 cable, the cost difference for Cat-5 is insignificant when compared to the installation cost, etc. When you consider that Cat-5 can be used for 100Base-T (and perhaps faster) systems, the choice is pretty clear.

Do not, in any case, attempt to use ordinary telephone cable (or other wiring you may have available) for your Ethernet runs. Cat-5 cable is state of the art stuff, employing carefully-controlled insulation, twist rates, etc. As a result, both crosstalk (between the pairs) and radiation (to outside receivers) is minimized.

You can, however, get away with running two 10Base-T connections over a single cable. Both Cat-3 and -5 have four pairs, but 10Base-T only needs two of them. 100Base-T requires all four pairs, however, so this may not be a good long-term solution.

If, when you were wiring up your network, you had the wisdom to use Cat-5 cable, you are in a great position to upgrade to 100Base-T. Because of the hub-and-spoke topology, all you need to do is stuff a 100Base-T hub alongside your existing 10Base-T hub in the wiring closet (or wherever :-). Hook the machines with fast interfaces (e.g., iMacs) onto the new network; leave the slow ones on the 10Base-T network (or add 100Base-T cards)...

Looking ahead, a technology called Gigabit Ethernet is just around the corner. A version (1000Base-T) is being made to work with Cat-5 cable, so again you can will be able to upgrade without rewiring your network. Apple hasn't bought into Gigabit Ethernet yet, but we strongly suspect that they will, as soon as prices drop and availability increases.

IrDA

The Infrared Data Association standard (IrDA) is a serial communication protocol which is built on the conventions of Radio Standard 232 (RS-232), but uses infrared light rather than electrical connections. Where RS-232 is a non-return-to-zero (NRZ) format, IrDA uses short pulses of light. A one is represented by a pulse 3/16 of a bit period long and a zero is represented by the absence of a pulse at a specified time.

The communication channel is point to point, with no broadcast capability. Each point must have one IR transmitter (usually a light emitting diode (LED)) and a detector (usually a phototransistor). These devices are commonly installed as a pair, using a single package which includes a driver and preamplifier. Sometimes another chip is used to convert NRZ into the short pulses for transmission and to convert them back to NRZ for use by the serial port of a microprocessor.

The IrDA specification calls for a maximum path of a meter or so in the presence of normal room illumination, but there is no reason that longer ranges cannot be obtained with lenses. IrDA does not require careful pointing of communicating devices toward one another. Megabit per second speeds are possible; the lowest of the common RS-232 speeds is 2400 baud.

It is important not to confuse IrDA with the common IR television control, which uses a higher frequency carrier to modulate the light and thus achieve much longer range in the presence of room illumination.

USB

Intel's Universal Serial Bus (USB) is designed as a multi-point replacement for RS-232, allowing up to 127 devices to be connected. In the Apple world, it is also a replacement for the Apple Desktop Bus. Cabling consists of a (two conductor, twisted) shielded pair for bi-directional data, plus two heavier wires for five-volt power and ground. USB supports two speeds, 1.5 MHz and 12 MHz, which can exist together on the same bus. Wiring for 1.5 MHz need not be shielded; neither speed requires termination.

Two new connectors have been developed for use with USB. One has four pins in a row and is used for connecting devices to hubs. The other has a square shape and is used for connecting hubs to computers or other hubs. You may never see the square shaped one, because hubs often have captive cables or are built into things like keyboards.

The overall connection scheme is described as "tiered hub and spoke". The tier starts with a hub to which a few devices can be connected. When the top level runs out of connector space, another hub can be added (much as you would plug extension cords into a power outlet). Apple's USB keyboard is a hub, typically used to support a mouse.

The hubs need to have some intelligence, because they are required to sense the addition of a device to an operating bus and inform the host. Actually, the hub will not apply bus power to the device until the host has assigned an address. The bus also must recognize high speed communication by the host and not send it out over unshielded wires or to devices it knows can handle only the lower speed.

USB packets consist of a synchronization pattern, a frame identifier, a device address of seven bits (128 devices), a frame number, up to 1023 bytes of data, and a cyclic redundancy check (CRC) code.

The digital encoding format is "non return to zero inverted - mark" (NRZM), in which a change in the bit stream as transmitted represents a zero, while no change represents a one. This scheme uses minimal bandwidth, but it can be fouled up by a string of one bits which produce no transitions for synchronizing clocks. The solution is bit stuffing, in which the transmitter counts consecutive ones and inserts a zero after the sixth one. The receiver is charged with reversing the process.

USB's signal amplitude is low: 0.4 volts differential centered at 1.75 volts DC. As busses mature and electronic devices improve, lower voltages and currents can be used, requiring less signal power and creating less interference. Thus, serial communications have moved from current-loop devices (e.g., old teletypes) through the 12V RS-232 standard to USB's 1.75 volts.

Power management on the USB can be problematical. The hub is supposed to supply power for the bus, but individual devices (if sufficiently power-hungry) or collections of devices (if sufficiently numerous) can overtax the hub's capabilities. As a result, the supply voltage can be depressed to the point that the entire bus fails to operate.

Current limiting by the hub helps, but this is not a perfect solution. Self-powered hubs, which do not draw excessive power from the host computer, can also help. Fundamentally, however, it is up to users to keep track of the load they are demanding of their hubs, just as they would for electrical devices on a common circuit breaker.

Devices which derive their own power from another source must still connect to bus power, because of the signaling features involved with connection to a hot bus (high- and low-speed devices are identified by details of the power connection). A device which is plugged into the bus, but is otherwise in an off state, must still provide information to the host.

The USB protocol allows for priority handling of a channel. This is necessary for time-sensitive data, such as digitized audio. Priority channels are isochronous (allowing guaranteed real-time performance) and unidirectional. Blocked channels are bi-directional and may or may not include error checking.

Blocked data can be given a back seat by software, up to a limit in which the priority channel uses all available bandwidth. It is interesting that the G4 will provide two USB channels to avoid this sort of saturation.

Converters (e.g., to ADB, Ethernet, LocalTalk, and SCSI) are conceivable and most are already available. These often exhibit speed mismatches, however, and should be considered only as temporary solutions.

IEEE-1394

IEEE-1394 is a high-speed serial connection, developed by Apple and later codified by the IEEE specification. Apple markets the interface as FireWire; Sony markets it as i.LINK. It currently runs from 12.5 to 400 Mb/sec and higher speeds are forthcoming.

FireWire has been adopted for real-time video use by camera manufacturers and it is rapidly becoming the interconnection means of choice for external disk drives for which cabling is a consideration. Dedicated game computers (e.g., by Nintendo) have been a testing ground for FireWire concepts and hardware.

The FireWire cable consists of two twisted and shielded pairs and two power wires fitted into a special connector. There are no terminators - at least none that users have to be concerned with. FireWire devices need to have two connectors, one for the cable to a source and another for connecting the next device to the chain. The pins in the connectors are designed to allow hot plugging without damage even if power is on. Of course, interrupting a chain to add another device will affect the remainder of the chain. Repeaters, splitters, and bridges allow for free-form chaining of devices in stars and chains.

Unlike SCSI, FireWire arrived on the scene with a standardized connector, so we have some hope of reasonable long-term sanity in this matter. Unfortunately, the "standard" FireWire connector has already been compromised: video cameras use a 4-wire version of the cable with no power lines and you may need an adapter cable. Sigh.

The FireWire protocol is a peer-to-peer system which has the potential of connecting computers to each other as a high-speed AppleTalk net, but no hardware is yet available to support the feature. Work is also being done on an IP definition for FireWire and Jini (Sun's Java-based communication protocol) also seems like an interesting possibility; stay tuned...

Because FireWire is a peer-to-peer system, the distinction between computers and peripherals can become a bit blurry. It is conceivable, for instance, that a video camera could take control of a Macintosh, telling it how to process and store incoming data. Provision is made for asynchronous and isochronous communication for bulk transfers and time-critical one-way transfers, respectively, in a manner quite like the USB (but much faster :-).

AirPort

Apple's AirPort system is based on Lucent's WaveLAN implementation of the IEEE 802.11 Direct Sequence Spread Spectrum (DSSS) radio standard in the 2.4 GHz band. It is currently being shipped with the iBook, but its promise extends far beyond that use. Many sites may soon be able to run their entire networks on AirPort, avoiding the need to run cables.

Nonetheless, AirPort has some significant limitations. The primary one is speed: ten Mbps was fine for the 90's, but the client-server and multimedia needs of the coming decade will demand 100 Mbps (or even faster) links. Worse, like simple Ethernet hubs, each AirPort hub can only handle one packet at a time. So, the hub's speed is divided by the number of currently active sessions.

Although more hubs can be added, logistics may become a problem. Wired networks have no problem with tightly-packed sets of offices; the signals only go where the wires allow. Radio networks, in contrast, can have problems with distances, shielding, interference, asymmetric communication paths, etc.

We are thus concerned that AirPort may not handle large sites (e.g., offices and schools) as well as users might wish. Until a few positive reports of large-scale installations have come in, we would advise users to be cautious about building their plans around this technology.

AirPort uses direct sequence modulation spread spectrum technology. This technique lets users share a common frequency band without significant interference, though the "noise floor" does rise as more transmitters become active.

Specifically, each transmitter encodes its message, using a 40-bit, pseudo-random code. Receivers correlate the incoming data stream with a copy of a specific, desired code. This allows them to suppress other codes, which (are supposed to) appear like random noise. The packets are also encrypted and sequenced, making interception and spoofing very difficult.

Lucent contends that this makes AirPort as secure as a wired network; that is, the degree of difficulty one would have in cracking it is equivalent to the difficulty of gaining access to a physical network. We would respond that some physical networks are far better protected than others, so the comparison is bound to be a bit sketchy. In any event, we wouldn't advise you to rely on AirPort's security against nosy three-letter agencies or well-heeled competitors.

There is also a question regarding the "noise floor". AirPort's encoding technique is known as code division multiple access (CDMA). CDMA allows convenient and informal bandwidth sharing (without control of a telephone company switching channels or an FCC assigning frequencies), but it does not create bandwidth where there was none before.

As more users enter the unlicensed band (and they need not be AirPort users), the apparent noise level in a given locale will increase and the band can saturate. We simply cannot predict when problems will arise without the large scale test of millions of links in an environment set up by average users. Stand by.

Dodo Birds

We have ignored a few connections which were rarely used or have fallen into disuse; here are some scattered comments which you may find interesting.

There were several implementations of a Processor Direct Slot (PDS), which appeared in the SE and the II si among other less expensive Macs. They are a simple connection to the pins of the microprocessor, with the additions of power and ground lines. Ethernet ports, additional video, and modems were available on the market.

A special audio-visual interface connection has been provided in some Macs - especially those with built in support for digital signal processors (DSP). It was usually used in conjunction with a NuBus or PCI card to supply high quality audio.

Early Macs had a special 19-pin subminiature D connector, for use with external floppy drives.

A specialized version of the serial port, the geoport, was briefly used for connection to telephone lines with only a certified data access arrangement (DAA) connection. Using DSP, it was possible to program the functions of an external modem in software.

In Closing

Apple's early connection strategies were dominated by the need to minimize hardware components (and thus, production costs). So, for instance, none of the early Macs used Direct Memory Access (DMA) chips, relying instead on the CPU to transfer data between devices and mamory. As interface chips have become more economical, however, Apple has moved to more flexible and powerful interfaces.

Apple has also started to adopt and promote industry standards, a move which we greatly applaud. Although you may still need to download drivers from a web site, many of the new IEEE 1394, PC card, PCI bus, and USB devices should work with Macs. This should increase users' options in add-on peripherals, while decreasing the cost of such additions.

Interface Name	A/D, S/P	Physical Connection	Bus Topology	Distance(ft.); Speed (Mbps)	Fanout Limit	Common and Typical Uses
10Base-2	D, S	cable	shared bus	600; 10	none	med-speed networking
10Base-T	D, S	cable	hub/spoke	300; 10	none	med-speed networking
100Base-T	D, S	cable	hub/spoke	300; 100	none	high-speed networking

ADB	D, S	cable	shared bus	typ. 10; 0.01	16	low-speed serial devices
AirPort	D, S	radio	shared bus	150; 11	none	med-speed networking
Audio	A, S	cable	single link	typ. 10; 22 kHz	none	powered speakers, etc.
EIDE	D, P	cable	shared bus	1.5; 32-100	2	hard disk drives
FireWire	D, P	cable, socket	hub/spoke	13; 12.5 - 400	63	block devices
IrDA	D, S	infrared	single link	3; 4	1	med-speed serial devices
LocalTalk	D, S	cable	shared bus	1000; 0.2	254	low-speed networking
NuBus	D, P	socket	shared bus	0; 300	typ. 3-6	interface cards
PC Card	D, P	socket	shared bus	0; 4 - 1000	typ. 1-4	interface cards
PCI	D, P	socket	shared bus	0; 500 - 2000	typ. 3-6	interface cards
RS-422	D, S	cable	single link	3000; 0.2	1	low-speed serial devices
SCSI	D, P	cable	shared bus	20-60; 80-600	7	block devices
USB	D, S	cable	hub/spoke	16; 1.5 - 12	127	med-speed serial devices
Video	A, P	cable	single link	typ. 10; varies	typ. 1	visual displays

Apple's IO Kit promises to add valuable features (e.g., strong multimedia support and hot-plugging) to the Mach/BSD underpinnings of Mac OS X. Thus, users with strong interests and needs for fancy I/O should keep a close eye on Mac OS X dev elopments over the coming year.

Note: many of the figures in the table above vary with implementations. Consequently, they are intended only for rough comparisons. For specific information, contact the vendor(s).

References and Links

AirPort
<http://www.apple.com/airport>
<http://www.apple.com/ibook/airport.html>
<http://www.wavelan.com/>

Ethernet
<http://www.ganymedesoftware.com/html/ethernet.htm>
<http://www.gigabit-ethernet.org/>

IEEE-1394 (aka FireWire, i.LINK)
<http://developer.apple.com/dev/firewire>
<http://www.1394ta.org>
<http://www.adaptec.com/technology/briefs/1394dif.html>
<http://www.skipstone.com/compcon.html>

IrDA (Infrared Data Association)
<ftp://irda.org>
<http://www.irda.org>

LocalTalk
<http://www.yale.edu/macguide/Trouble/Localtalk.html>

PC Card (aka PCMCIA)
<http://www.pc-card.com/pccardstandard.htm>
<http://www.pcmcia.org>

PCI (Peripheral Component Interconnect)
<http://www.pcisig.com>

USB (Universal Serial Bus)
<http://www.apple.com/usb/>
<http://www.ariston.com>
<http://www.themacintoshguy.com/lists/MacUSBFAQ.shtml>
<http://www.usb.org/>

A 30-year veteran of the computer industry, **Rich Morin** <rdm@ptf.com> writes the Silicon Carny column for SunWorld magazine and is a Contributing Editor for MacTech. He programs almost entirely in Perl these days, on Mac and Unixish systems. Rich is also the president of Prime Time Freeware (http://www.ptf.com), which publishes mixed-media (book/CD-ROM) collections of freely redistributable software. PTF's Mac-specific products include "MacPerl: Power and Ease" and "MkLinux: Microkernel Linux for the Power Macintosh".

Doug P. McNutt <dmcnutt@macnauchtan.com> earned the PhD degree in physics from the University of Wisconsin and spent most of his life doing space research at the U.S. Naval Research Laboratory where electronic techniques, including computing, were a means to an end. He is currently a consultant involved with applications of the Apple Macintosh and smaller microprocessors (e.g., for remote measurement of quality of electrical power).