

# The IrDA Standards for High-Speed Infrared Communications

Iain Millar

Martin Beale

Bryan J. Donoghue

Kirk W. Lindstrom

Stuart Williams

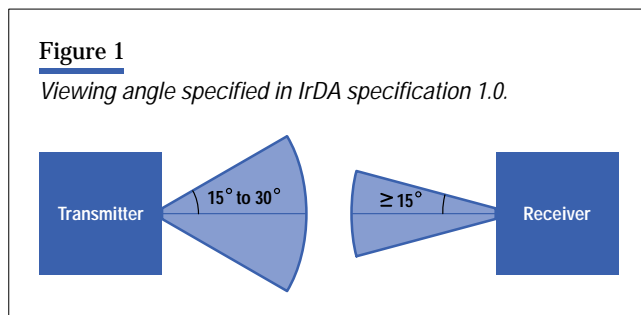
As more data communications products, such as printers and laptop PCs, are released with infrared capability, support for a core set of IrDA standards has strong support from many manufacturers because, among other things, they want to ensure that their products will interoperate in a transparent and user-friendly manner.

**T**he use of infrared techniques for data communications has been around for several years, and by 1993 several commercial products were available with this capability. However, each company has tended to have its own infrared standard, and although devices from the same manufacturer could communicate with each other, competing systems tended not to be interoperable. Examples of such proprietary infrared systems include Hewlett-Packard's HP SIR (serial infrared), Sharp's ASK systems, and General Magic's MagicBeam. The resulting confusion in the marketplace meant that users viewed infrared as having only limited utility.

On June 28, 1993, the Infrared Data Association (IrDA) had its first meeting with the purpose of establishing a ubiquitous, low-cost, point-to-point serial infrared standard. Some 50 representatives from 20 interested companies were expected, but over 120 people representing more than 50 companies actually attended. It was clear that the industry was interested in developing a standard that would allow the true value and utility of infrared to be realized. At the culmination of this process—and due in no small part to the enthusiasm and spirit of cooperation of the participating companies—the first IrDA standards were published, just one year and two days after the initial meeting.

To date, IrDA has specified the physical and protocol layers necessary for any two devices that conform to the IrDA standards to detect each other and exchange data. The initial IrDA 1.0 specification detailed a serial, half-duplex, asynchronous system with transfer rates of 2400 bits/s to 115,200 bits/s at a

range of up to one meter with a viewing half-angle of between 15 and 30 degrees (see **Figure 1**). More recently, IrDA has extended the physical layer specification to allow data communications at transfer rates up to 4 Mbits/s.



This paper presents details of the individual IrDA specifications, focusing specifically on the high-speed extensions that allow data communications at up to 4 Mbits/s. The first section gives details of the objectives that resulted in the series of IrDA specifications. The specifics of the user model and the technical requirements of the specification are also presented. Next the IrDA architecture is introduced, highlighting how the IrDA specifications together provide overall functionality. The infrared physical-layer specification with particular emphasis on modulation format, packet framing, transceiver design, and clock recovery is discussed in the next section. The transceiver design for the HP HDSL-1100 IrDA transceiver is also described in this section. The last section covers the protocol layers of the IrDA specifications. Finally, IrDA's current status is summarized.

#### IrDA Objectives

When IrDA was established, it set for itself the following objective:

"To create an interoperable, low-cost infrared data interconnection standard that supports a walk-up, point-to-point user model\* that is adaptable to a broad range of mobile appliances that need to connect to peripheral devices and hosts."<sup>1</sup>

IrDA chose the short-range, walk-up, point-and-shoot directed infrared communications model for two main reasons. First, it was perceived that the initial target market for IrDA-enabled devices would be the mobile

professional who is also a computer user. The environment for the use of such devices would be in a typical working environment in which the majority of stationary devices, such as printers or computers, would be located within the user's own reach space, on the desktop or in the immediate vicinity. Typical use of such devices would consist of short, conscious interactions such as file transfer or printing. Such use scenarios do not require the devices to be continually connected to each other, and a directed model of communications was adopted in which the user consciously points the infrared device at the target.

Previously, mobile professionals might connect their laptops to various peripherals using parallel or serial cables. Connecting such devices using LAN connections might also be possible if cost were not an issue. However, a problem arises when the user becomes mobile—for example, when visiting customers in their office. Setting up a laptop at the customer office to achieve even simple tasks, such as printing or file transfer, would more than likely require significant reconfiguration. IrDA aimed to change this by providing standards for ubiquitous access to such devices.

Second, IrDA chose this communications model to minimize cost. The use of a single LED and photodiode in the transceiver enables an extremely low-cost implementation. The model simplifies the protocol software by restricting the number of visible devices, hence limiting the contention and interference between IrDA devices. The limited range also allows reuse of the infrared medium, allowing multiple pairs of devices to communicate at the same time.

\* The phrase "walk-up, point-to-point user model" refers to the fact that to ensure data transfer between devices with infrared capabilities, they must be placed close together (< 2 m) with their infrared transceivers pointed at one another.

## Glossary

**Cell.** A symbol in PPM.

**Chip.** A pulse within a symbol (cell) in PPM.

**ENDEC.** The encoder-decoder used in the IrDA physical layer.

**HTTP** Hypertext Transfer Protocol.

**HDLC.** A bit-oriented, synchronous High-level Data Link Control protocol that applies to the message-passing (data link) layer of the Open Systems Interconnect (OSI) model for computer-to-computer communications.

**IAS.** The information access service maintains information about the services available on the host device and provides services that allow access to information on remote devices.

**IrCOMM.** IrDA specification for the emulation of serial and parallel port communications.

**IrLAN.** IrDA specification for accessing a LAN over an infrared medium.

**IrLAP.** IrDA specification for Link Access Protocol. This document specifies an HDLC-based protocol for controlling access to the infrared medium.

**IrLMP.** IrDA specification for Link Management Protocol. This protocol provides the LM-MUX and LM-IAS services.

**IrOBEX.** IrDA specification that defines the protocol for generic object exchange in an IrDA-enabled device.

**IrPHY.** The specification that describes the physical layer properties of the IrDA standard.

**LM-IAS.** The Link Management Information Access Service allows a pair of IrDA devices to interrogate each other to determine the services available on each device.

**LM-MUX.** The Link Management Multiplexer allows any pair of IrDA devices to simultaneously and independently use a single IrDA connection between themselves.

**LSAP.** Link Service Access Ports are address fields that uniquely identify applications on the source and destination devices.

**LSAP-SEL.** Link Service Access Port Selector.

**PPM.** Pulse position modulation.

**SIR.** Serial infrared.

**Tiny TP.** Lightweight transport protocol specification.

IrDA aimed to allow its standards to support a wide class of computing devices and peripherals that might be used by mobile professionals. These devices would range from very sophisticated, high-power notebook or laptop personal computers, through palmtop computers and personal digital assistants, to simple single-function devices like electronic business cards or phone dialers. Target peripheral devices would include conventional computer-oriented devices like printers and modems, as well as automatic teller machines and public and mobile telephones. It was also envisaged that IrDA would enable new classes of devices such as information access points.

To target such a broad range of devices, a set of general requirements was placed on any prospective standard. These requirements included:

- Low cost
- Industry standard
- Compact, lightweight, low-power

- Intuitive and easy to use

- Noninterfering.

Using these requirements, the IrDA committee developed a series of standards aimed at providing ubiquitous, low-cost, directed infrared communications for all classes of mobile computing devices. In IrDA's vision of the world, the user of such devices would be able to roam across international boundaries using IrDA communications to access information, computing, and communications services in a uniform and transparent manner. The days of the mobile computer user travelling the globe with a multitude of modem, serial, and parallel cables, including adapters, will be gone.

The remainder of this paper presents details of the standard IrDA has put in place to achieve this vision.

#### The IrDA Architecture

After the initial marketing requirements had been specified, the technical committee within IrDA moved quickly towards the development of the initial standards. In April 1994, the first IrDA standard was published covering the physical layer properties. This document, the Infrared Physical Layer (IrPHY) specification,<sup>2</sup> describes an infrared transmission system based on a UART modulation strategy. The document specifies the necessary parameters to provide an asynchronous half-duplex serial communications link over distances of at least one meter at data rates between 2400 bits/s and 115.2 kbits/s. The cone half-angle of the infrared transmission is specified as being at least 15 degrees, but no more than 30 degrees. The IrPHY specification was quickly followed with the publication of the Infrared Link Access Protocol (IrLAP) in June 1994.<sup>3</sup> IrLAP specifies an HDLC-based protocol for controlling access to the infrared medium and providing the basic link-level connection between a pair of devices.

During the development of IrPHY and IrLAP, it was realized that some additional functionality was required in addition to the ability to provide a single connection between a pair of devices. The Infrared Link Management (IrLMP) layer was conceived.<sup>4</sup> This layer has two primary functions.

First, it provides the mechanism by which multiple entities within any pair of IrDA devices can simultaneously and independently use the single IrLAP connection between those devices. This function is called the link management multiplexer (LM-MUX).

Second, it provides a way for entities using the IrDA services to discover what services are offered by a peer device and to register available services within the local device. This link management information access service (LM-IAS) considerably benefits the ease of use of portable devices, allowing pairs of devices to interrogate each other to discover information about the applications within each device.

These three standards—IrPHY, IrLAP, and IrLMP—form the core of the IrDA architecture, and all are required for a device to be IrDA-compliant. Since the core documents were published, several extensions have been added. The current complete IrDA architecture is shown in **Figure 2**.

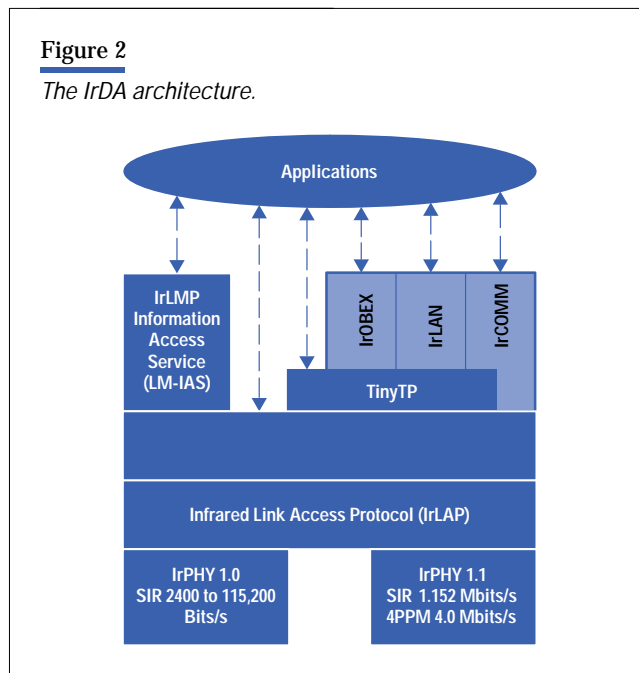
In October 1995, optional extensions to the physical layer, adding data transmission speeds of up to 4 Mbits/s, were accepted by the IrDA committee. These changes resulted in the IrDA IrPHY 1.1 specification.<sup>5</sup> The IrLAP and IrLMP documents have also recently been updated to version 1.1 to incorporate various improvements that resulted from practical experience in implementing and using the IrDA protocols.<sup>6,7</sup>

In addition to the base standards, IrDA has specified a protocol called Tiny TP.<sup>8</sup> This protocol is an extremely lightweight transport protocol designed to provide application-level flow control as well as segmentation and reassembly of application data units. This protocol has proved to be useful and is now implemented by most applications that support the IrDA architecture.

To complement the functionality of the main components of the IrDA architecture, several application-level protocols have been and are in the process of being developed. These protocols are aimed at providing convenient and uniform interfaces to the functionality of the IrDA protocols for both old and new applications.

**Figure 2**

*The IrDA architecture.*



The original target for IrDA was cable replacement. The need for a protocol to support the redirection of serial and parallel cable traffic resulted in the IrCOMM serial and parallel port emulation protocol specification.<sup>9</sup> This protocol enabled the redirection of conventional serial and parallel ports over the infrared medium, allowing many existing applications to operate unchanged over an IrDA link. Another area seen as a suitable application of IrDA, particularly as a result of the high-speed extensions, is wireless access to local area networks. The protocol IrLAN was developed to allow an IrDA-enabled device to access a LAN over the infrared medium.<sup>10</sup> The protocol, in combination with an IrLAN-compatible LAN access device, provides the IrDA device with the equivalent functionality of a LAN card and the advantages of wireless connectivity.

Both IrCOMM and IrLAN address legacy-style applications. However, it is envisioned that many new applications will be enabled by the IrDA standards. Using IrDA on low-end devices gives rise to the need for a flexible, lightweight information exchange protocol suitable for devices with varying resource capabilities. A protocol for generic object exchange, IrOBEX, is currently under development within IrDA.<sup>11</sup> This protocol is based on HTTP (Hypertext Transfer Protocol) but is more compact. When completed, IrOBEX will provide a device independent method for exchanging arbitrary units of data between IrDA-enabled devices.

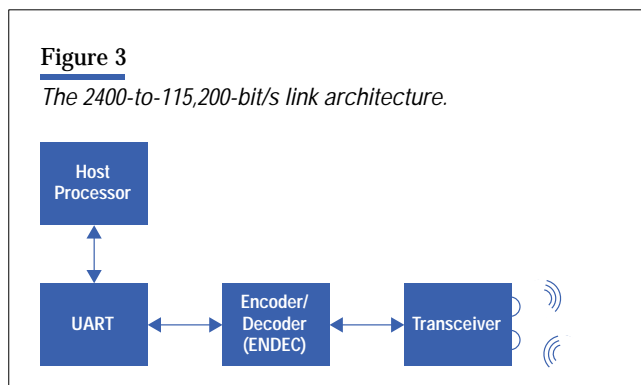
## The IrDA Physical Layer

The IrDA physical layer is split into three distinct data rate ranges: 2400 to 115,200 bits/s, 1.152 Mbits/s, and 4 Mbits/s. Initial protocol negotiation takes place at 9600 bits/s, making this data rate compulsory. All other rates are optional and can be added if a device requires a higher data rate. The links are designed to be used in a line-of-sight, point-and-shoot manner and hence have a modest minimum coverage of one meter, with a  $\pm 15^\circ$  viewing angle. This modest coverage is advantageous, since it allows a low-cost, high-data-rate link to be produced in a small package.

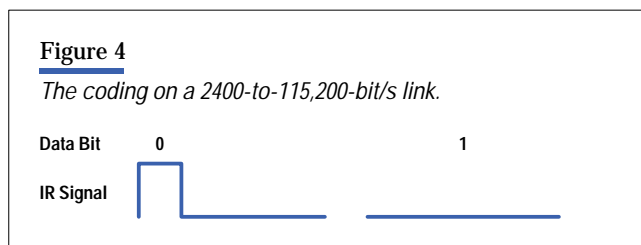
### 2400-to-115,200-bit/s Link

This is based on the HP-SIR link developed for HP calculators.<sup>12</sup> All IrDA-compliant devices implement this type of link since initial protocol negotiation takes place at 9600 bits/s. The architecture of the link (**Figure 3**) is designed for easy implementation and low cost. Hardware costs can be kept to a minimum by implementing the protocol, packet framing,

and CRC calculation in software on the host processor. Bytes of data from the processor are converted to a serial data stream by a UART (universal asynchronous receiver-transmitter). Since many systems already include a UART for RS-232 communications, this places no extra cost burden on the system. Only the ENDEC (encoder-decoder) and transceiver represent an additional hardware cost for the system.



Infrared receivers contain a high-pass filter to remove background daylight. This high-pass filter forces the use of encoding on the link to ensure that long strings of zeros or ones are not lost in transmission. The encoding used on this link is return-to-zero (RZ). Zeros are represented by a pulse of 3/16-bit duration, and ones by the absence of a pulse (**Figure 4**). For example, 3/16 of a pulse width at 115,200 bits/s is 1.6  $\mu$ s. The code is power-efficient since infrared light is only transmitted for zeros. The tall narrow pulse has better signal-to-noise ratio performance than a short wide pulse of the same energy.



#### The 1.152-Mbit/s Link

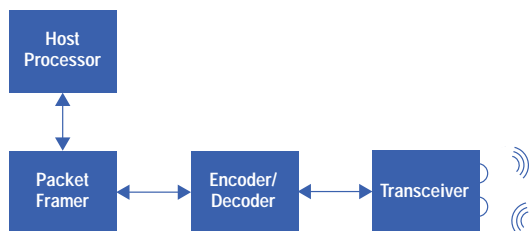
At speeds above 115,200 bits/s, packet framing and CRC generation and checking become a significant burden to the host processor. At 1.152 Mbits/s, these tasks are performed in hardware by a packet framer (see **Figure 5**). The packet format is slightly different from that used in the 2,400-to-115,200-bit/s link, but the line code remains similar.<sup>5</sup> Higher-level protocols are less processor intensive than packet framing or CRC generation and are still implemented in software on the host processor.

#### The 4-Mbit/s Link

The 4-Mbit/s link architecture is shown in **Figure 6**. As in the 1.152-Mbit/s link, packet framing and CRC generation and checking are performed in hardware to relieve the burden on the host processor, while higher-level protocols are implemented in software on the host processor. The link uses a new encoding scheme (described below) and a new, more robust packet structure. A phase-locked loop replaces edge detection as the means of recovering the sampling clock from the received signal. The packet framer, ENDEC, and phase-locked loop are more complex than the UART and ENDEC in the 2400-to-115,200 bit/s link. However, this added complexity need not be expensive. The components are specified in a

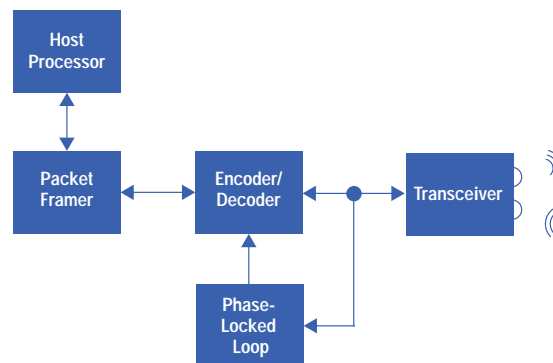
**Figure 5**

*The 1.152-Mbit/s link architecture.*



**Figure 6**

*4-Mbit/s link architecture.*

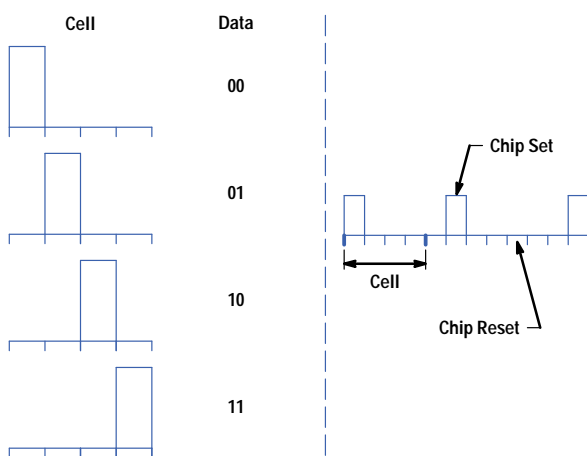


hardware description language and can be added quickly and inexpensively to one of the host system's ASICs. PC chip-sets including the 4-Mbit/s hardware are already available from leading semiconductor manufacturers.

**Coding and Packet Format.** Pulse position modulation (PPM) was chosen as the line code for the 4-Mbit/s link. Data is transmitted within a PPM signal by varying the position of a pulse (referred to here as a chip) within a symbol (referred to here as a cell). The PPM modulation for the 4-Mbit/s link allows one chip to be set in one of four possible positions; thus it is known as 4PPM. Since a chip can be set in one of four possible positions, four different messages can be sent within one cell, allowing two bits of data to be encoded per cell. **Figure 7** shows the four possible messages that can be transmitted by 4PPM.

**Figure 7**

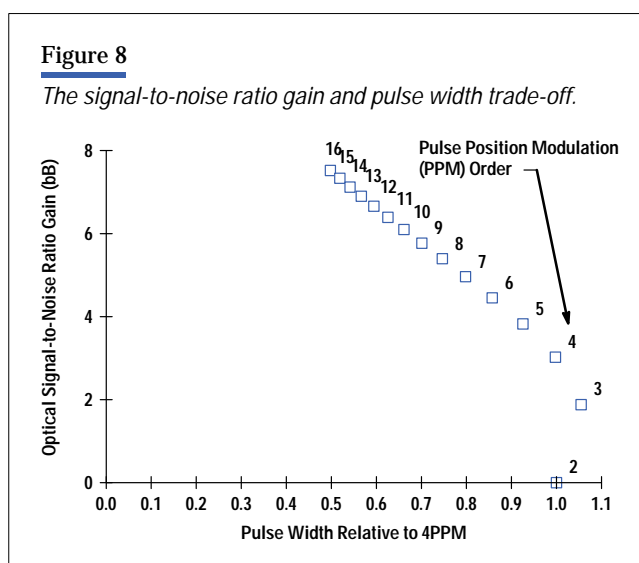
*4PPM message encoding.*



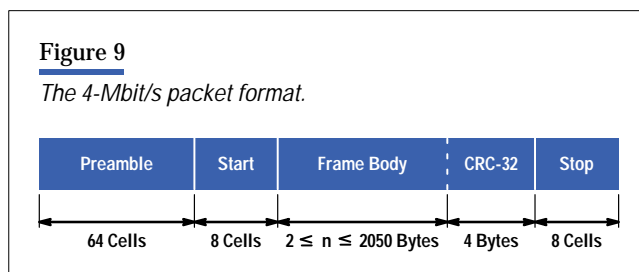
Pulse position modulation has many properties that make it attractive for use on the free-space optical channel. One of the main properties is the sparseness of the code. Sparse code allows high peak powers to be employed for set chips while maintaining a reasonable average power. The eye-safety rules stipulate a maximum average optical power, and LEDs tend to be average-power-limited at moderate duty cycles.

Pulse position modulation also contains significant and regular timing content, which facilitates synchronous clock recovery using a phase-locked loop. It is a modulation format that has very little dc content and can be high-pass filtered at 100 kHz, avoiding interference generated by fluorescent lighting without adversely affecting the receiver's eye diagram. A particularly interesting feature of PPM—one that had important ramifications in the choice of end delimiters—is its ability to detect line code errors.

Higher orders of PPM give lower duty cycles and theoretically greater signal-to-noise ratio gains on the infrared medium. **Figure 8** illustrates the interesting relationship between signal-to-noise ratio gain achievable with various orders of PPM and the required pulse width. It is interesting to note that the optimum order of PPM from a bandwidth efficiency perspective would be 3PPM. This result might be of theoretical interest, but is fairly useless in a practical system. Since the fastest bright LEDs have a rise time of around 40 ns, and the rise time of an LED is proportional to the pulse width, the use of high-order PPMs at 4 Mbits/s becomes impractical. The decision to adopt the order four for the PPM was motivated by knowledge of the range of duty cycles over which LEDs are peak-power-limited, the rise and fall time of available LEDs, and the frequent timing content provided at order four.



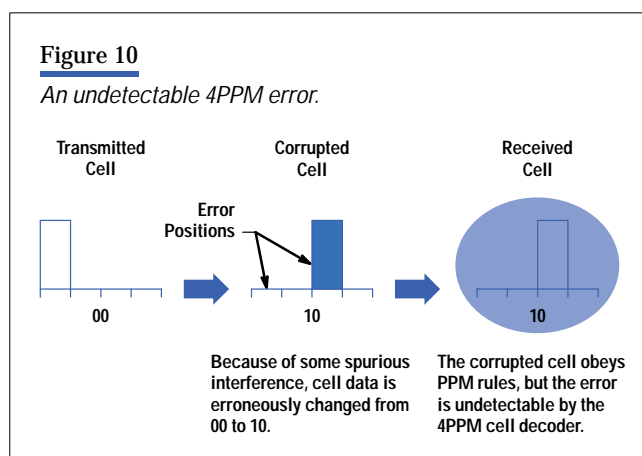
Packet Format. The 4-Mbit/s physical layer packet has distinct features that perform a useful and well-defined role (see **Figure 9**). A preamble allows dc balance to be attained, and more important, permits the phase-locked loop to achieve chip-level synchronization. The length of the preamble was considered carefully such that the preceding two goals could be achieved without a significant impact on efficiency. The start and stop delimiters provide cell and frame synchronization and were chosen so as not to compromise overall packet robustness or adversely affect the receiver eye diagram. To distinguish the preamble and the end delimiters from the frame body, these fields contain code violations. The body of





the packet is 4PPM-coded and has a 32-bit cyclic redundancy check (CRC) field appended to it. The choice of a 32-bit CRC provides a guaranteed level of robustness to undetected data errors over the range of error rates expected on a free-space infrared channel. The CRC is performed on the data bits rather than on the PPM-encoded chips.

**Error Detection and Delimiters.** A decoder may choose to exploit the error detection capabilities of 4PPM. The only portions of the packet allowed to contain violations are the preamble and the frame delimiters. If a decoder finds code violations within the frame body or CRC portion of the packet, it can flag that packet as being corrupted. In the same way that a sufficient number of carefully positioned errors can produce a correct-looking CRC for a corrupted packet, there are some error patterns that a 4PPM decoder cannot detect. An example is shown in **Figure 10**.

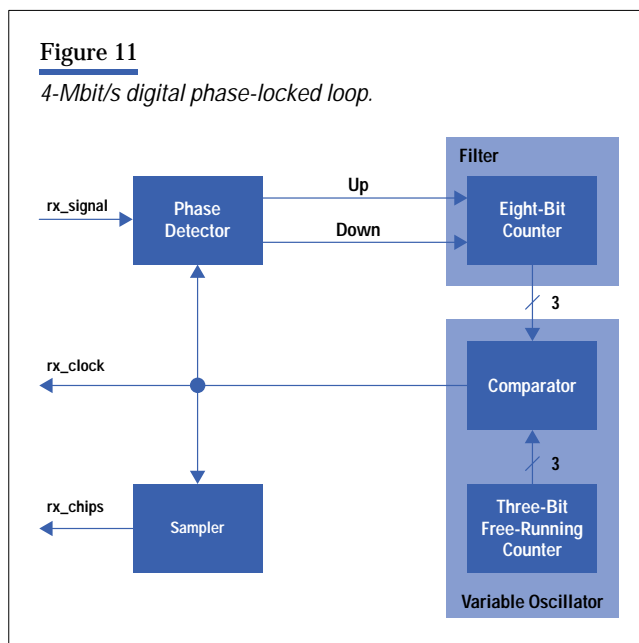


The role of the CRC is to detect those error patterns that the PPM cell decoder cannot detect. Owing to the combined distance structure of the CRC and the pulse position modulation, the packet can be made very robust to withstand either random or burst errors at any signal-to-noise ratio.

A more worrisome error mechanism that had to be considered was the possibility of the corruption of the frame delimiters. The frame delimiters are not in themselves protected by the CRC. If the situation arose whereby a false Stop delimiter appeared in a valid position within the data and CRC portion of the packet, the packet would be protected solely by the scrambling effect of the CRC. In this case, a corrupted packet would be flagged as correct with a probability of  $(0.5)^{32}$ . Thus, it is important to ensure the unlikelihood of either random or burst errors causing a false delimiter to appear within the data portion of the packet. This is achieved by choosing delimiters with a large Hamming distance from the data (or shifted versions of the data, to ensure serial uniqueness) and with a sufficient number of chips such that “bursty” channel error models can be tolerated. A further constraint on the delimiter choice is that delimiters must not adversely affect the eye diagram of the complete packet. The lack of long strings of contiguous set or reset chips within the 4-Mbit/s delimiters allows this goal to be attained. The delimiters chosen ensure packet robustness at any signal-to-noise ratio, for any length of packet, over random and burst-error models—all without affecting the receiver eye diagram.

**Clock Recovery.** The UART-style clock recovery of the 2400-to-115,200-bit/s link uses a single signal edge to set the phase of the recovered sampling clock. This inevitably gives rise to phase jitter on the recovered clock and a consequent signal-to-noise ratio penalty. The phase-locked loop used by the 4-Mbit/s link generates a sampling clock with much less jitter because it uses timing information from many signal edges to set the phase of the clock. An analog phase-locked loop could have been used for clock recovery and might have achieved a low phase jitter, but it would have been unable to achieve the rapid phase lock of a digital phase-locked loop. Rapid phase lock is important in a packetized data system, because it determines the length of the training sequence, or preamble, required at the start of every packet to allow the phase-locked loop to lock.

The lock time is dictated by the accuracy with which the nominal frequency of the phase-locked loop's variable oscillator can be set. The nominal frequency of the variable oscillator in an analog phase-locked loop is highly variable, since it is determined by the (usually poor) tolerance of the resistors and capacitors. By contrast, the nominal frequency of the variable oscillator in a digital phase-locked loop can be locked to a crystal reference with a tolerance of less than 100 ppm. Implementations of digital phase-locked loops have the additional advantage that they can be quickly and easily ported between ASIC designs. The architecture of a typical digital phase-locked loop for the 4-Mbit/s link is shown in **Figure 11**.



The phase detector is a state machine that compares the edges in the received signal ( $rx\_signal$ ) with those of the recovered clock ( $rx\_clock$ ). Rising edges only occur in  $rx\_signal$  at PPM chip boundaries. Rising edges of  $rx\_clock$  should occur halfway between chip cell boundaries. If  $rx\_signal$  is earlier than expected, then the phase detector produces a Down signal, thereby advancing the phase of  $rx\_clock$ . If  $rx\_signal$  is later than expected, then the phase detector produces an Up signal.

The three most significant bits of the 8-bit counter set the phase of  $rx\_clock$ . The five least significant bits ensure that the counter acts as a low-pass filter, since many Up and Down signals are required to change the phase of  $rx\_clock$ . The three-bit free-running counter and the comparator together act as a variable phase oscillator. All blocks within the phase-locked loop are clocked by the same system clock. The system clock can be either 40, 48, 56 or 64 MHz, the choice being set by the rollover point of the three most-significant bits of the 8- and 3-bit counters (100, 101, 110, or 111). A 40-MHz system clock means that  $rx\_clock$  should be very granular, with only five possible phase steps within a chip period. The effective number of phase steps is, however, doubled by making use of both the positive and negative edges of the system clock in the phase detector and sampler. The choice of whether to use positive or negative edges can be made by examining the fourth most-significant bit of the 8-bit counter.

The fast lock of the digital phase-locked loop is further aided by using a dual control loop within the digital phase-locked loop. A lock state machine within the phase detector decides whether the digital phase-locked loop is in or out of lock by examining the average deviation of the  $rx\_clock$  edges from the  $rx\_signal$  edges. If the digital phase-locked loop is out of lock, then multiple Up or Down pulses are generated for each edge in  $rx\_signal$  to ensure rapid lock. Once locked, only single Up or Down pulses are generated since multiple pulses would increase phase jitter on  $rx\_clock$ .

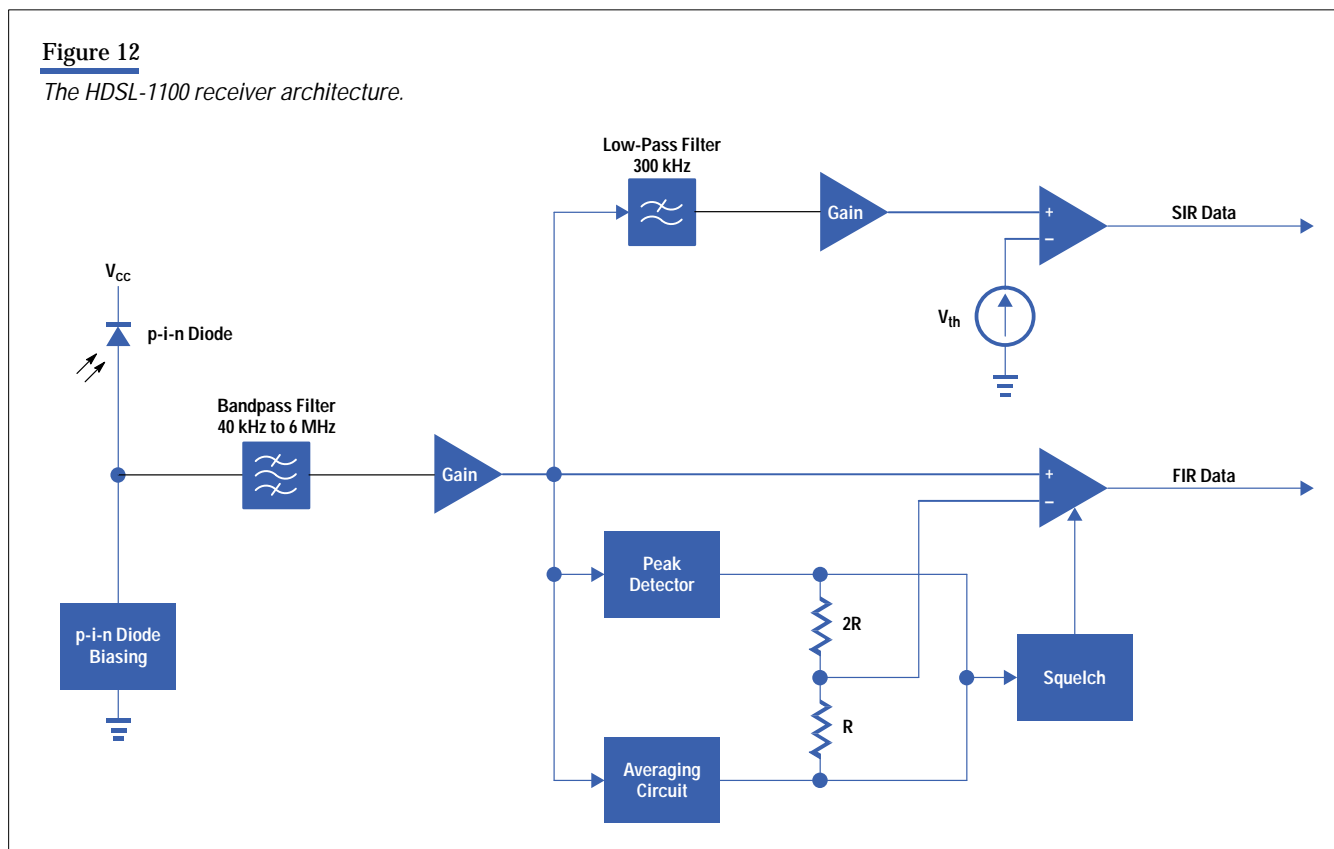
#### The Hewlett-Packard HSDL-1100 IrDA Transceiver

The HP HDSL-1100 from HP's Communication Semiconductor Solutions Division is the world's first fully IrDA-compliant transceiver capable of operating at all IrDA data rates from 2400 bits/s to 4 Mbits/s. The HSDL-1100 fits within the same small package as its predecessor, the HSDL-1000, which operated at data rates from 2400 bits/s to 115,200 bits/s. The small package size available for pins, IC, passive components, and heat dissipation imposed design constraints on the complexity of the transceiver. The IC uses a low-density bipolar in-house process, which is low in cost and allows quick turn times on wafers for IC development.

Transmitter design was straightforward. However, the multiple data rates, line codes, and large dynamic range made receiver design much more challenging. The receiver's dual-channel architecture is shown in **Figure 12**. A shared p-i-n diode detects all infrared signals with a modulation frequency between 40 kHz and 6 MHz. An amplifier boosts this signal before it is split into separate receiver channels. IrDA signals at 2400 to 115,200 bits/s pass through the serial infrared (SIR) channel\*, while 1.152-to-4-Mbit/s signals pass through the fast infrared (FIR) channel. The lower bandwidth of the SIR channel (40 to 300 kHz) means lower noise and allows the SIR channel to meet the IrDA  $4 \mu\text{W}/\text{cm}^2$  sensitivity requirement. The higher-bandwidth (40 kHz to 6 MHz) FIR channel has higher noise, but still meets the  $10 \mu\text{W}/\text{cm}^2$  sensitivity requirement for 1.152-to-4-Mbit/s IrDA links. Since the different data rate IrDA links overlap in their modulation spectra, the received signal will appear on both channels. The ENDEC relies on information provided by the protocol to ensure that it listens on the correct channel.

**Figure 12**

*The HDSL-1100 receiver architecture.*



\* At low rates, such as 2400 or 9600 baud, only the leading edge of the signal passes through the 40-kHz to 6-MHz bandpass filter. The signal is still correctly decoded since the ENDEC is able to tolerate received SIR pulses as short as 1 to 4  $\mu\text{s}$ .

The receiver converts signals from an analog to a digital form by comparing them with a threshold voltage. The two channels have different threshold detection circuits to meet the different requirements for the signals. The SIR channel has a fixed threshold set at the level of the weakest received signals. Although the fixed threshold tends to extend the duration of high-level pulses, the line code for the 2400-to-115,200-bit/s ENDEC is tolerant of pulses that extend to five times their nominal width. The 4-Mbit/s ENDEC is far less tolerant of pulse extension, so a dynamic threshold is required on the FIR channel. The dynamic threshold tracks the 50% level between the peak extensions of the 4PPM signal. A peak detector tracks the 100% level of the signal and an average circuit tracks the 25% level. The 50% threshold level is derived from a 2R-R voltage divider connected to these levels. Between packets, the dynamic threshold drops to zero. This would allow the FIR\_Data output to “chatter” on noise or on feedback between the output pin and the p-i-n diode. The 1.152-Mbits/s ENDEC is intolerant of the extra pulses produced by such chatter, so a squelch circuit was added to switch off the FIR\_Data output at low signal levels. The dynamic threshold also takes time to settle at the start of a packet, which causes some of the packet’s initial infrared pulses to be lost or distorted. While this would be disastrous for the 2400-to-115,200-bit/s link, the 1.152- and 4-Mbit/s packets include a preamble to allow the receiver to settle before decoding data.

Another challenge for receiver design was the dynamic range of infrared signals. IrDA specifications allow received signal strength to vary between  $4 \mu\text{W}/\text{cm}^2$  and  $500 \text{ mW}/\text{cm}^2$ . This is a dynamic range of 51 dB. Since the p-i-n diode is a square law detector, this dynamic range doubles to 102 dB within the receiver. The receiver achieves this dynamic range by allowing the signal to be clipped while maintaining the timing of the signal. The impedance of the p-i-n diode biasing circuit decreases with signal level, reducing the signal voltage and the receiver amplifier’s limit without saturating. The p-i-n diode has also been carefully designed to ensure that the induced signal decays rapidly once an infrared pulse disappears.

## The IrDA Protocol Layers

### The Infrared Link Access Protocol

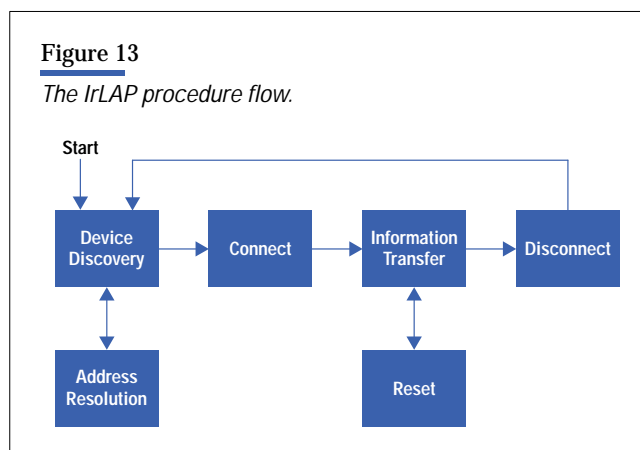
IrLAP is the IrDA protocol that provides the basic link layer connection between a pair of IrDA devices. It is based on the HDLC protocol providing functions like connection establishment, data transfer, and flow control.<sup>13,14</sup> However, IrLAP has significant additional features as a result of the specific properties of the infrared medium.

The infrared medium over which IrLAP is required to operate is a point-to-point, half-duplex medium. While the narrow cone angle of IrPHY limits the number of other devices that can be seen, it does increase the probability of hidden devices. In such a situation, one device may see many other devices. However, it does not follow that those devices will see each other. This can result in collisions where transmissions from devices hidden from each other may overlap, resulting in the inability of the receiving device to decode those frames correctly. The characteristics of the infrared medium also result in there being no reliable way to detect transmission collisions. Conventional carrier sensing with collision-detection protocols would therefore be unsuitable, and IrLAP provides a mechanism for ensuring contention-free access to the medium, at least during data transfer.

The IrLAP has three distinct phases of operation: link initialization, nonoperational mode, and operational mode. Nonoperational and operational modes are distinguished by the absence or presence of a connection with another device. During link initialization, the IrLAP layer chooses a random 32-bit device address. This address is randomly chosen to negate the need to select and maintain fixed device addresses for all IrDA devices. Although it is unlikely that two or more devices within range of each other will choose the same address, procedures are defined to detect and resolve address collisions. After the link is initialized, the IrLAP layer enters nonoperational mode.

The nonoperational mode is derived from HDLC's normal disconnect mode (NDM). In this mode, all devices contend for the medium. To do this, each device must check that the medium is not busy before transmission. This is achieved by listening for activity—that is, listening for physical layer transitions for at least 500 ms. Transmissions in the normal disconnect mode use link parameters that can be supported by all IrDA devices at a rate of 9600 bits/s. In this mode, the device will initiate device discovery, address resolution (if required), and connection establishment.

Once the connection has been established, the IrLAP layer moves into the operational or, in HDLC terms, normal response mode. This mode is an unbalanced mode of operation in which one device assumes the role of primary station and the other assumes a secondary role. This is the phase in which information is exchanged under control of the primary station. The link parameters are negotiated during the connection setup procedure and remain constant during the connection. During this phase, all other devices within range of either the primary or secondary stations remain idle in the normal disconnect mode. The two communicating devices therefore have unrestricted access to the medium for the duration of the connection. Once the information has been transferred, the link is disconnected and the device returns to the normal disconnect mode. The flow of procedures for the IrLAP layer is shown in **Figure 13**.



**Device Discovery and Address Resolution.** The discovery procedure is the process an IrDA device uses to determine whether or not there are any devices within communications range. In doing so, the device discovers the address of any device within range, the version number of the IrLAP protocol operating in each device, and some discovery information specified by the IrLMP layer in each device. The discovery procedure is controlled by the initiating device, which divides the discovery process into equal periods or time slots. The slotted nature of the discover procedure minimizes the likelihood of collisions when there are multiple devices within range.

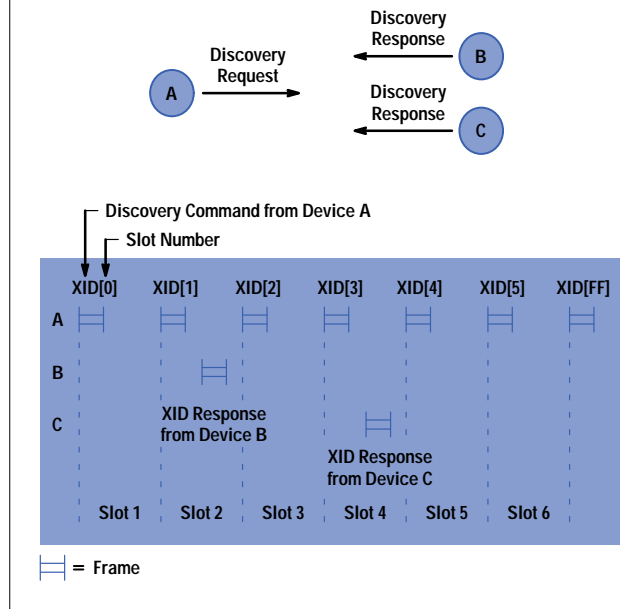
After waiting for a period of 500 ms (normal disconnect mode rules), the initiating device starts the discovery procedure and broadcasts frames marking the beginning of each slot. On hearing the initial discovery slot (which also details the number of slots in the discovery process: 1, 6, 8 or 16), a device randomly selects one of the slots in which it will respond. When the device receives the frame marking its chosen slot, it transmits a discovery response frame. All frames in the discovery procedure use the HDLC unnumbered format of type XID (exchange identification).<sup>\*</sup> An example of the discovery process is shown in **Figure 14**.

**Figure 14** shows a three-device scenario in which device A is within range of devices B and C. Device A initiates the discovery process by transmitting a discovery XID command frame which, in this case, indicates that this is a six-slot discovery process and that this is the initial slot. Device A continues to transmit discovery command XID frames indicating the appropriate slot number. The final frame, after slot 6, is indicated by a slot number 0xFF. The final slot also contains information about the initiating device.

<sup>\*</sup> In this context XID is a type of HDLC frame as specified in the ISO standard.

**Figure 14**

*The discovery procedure.*

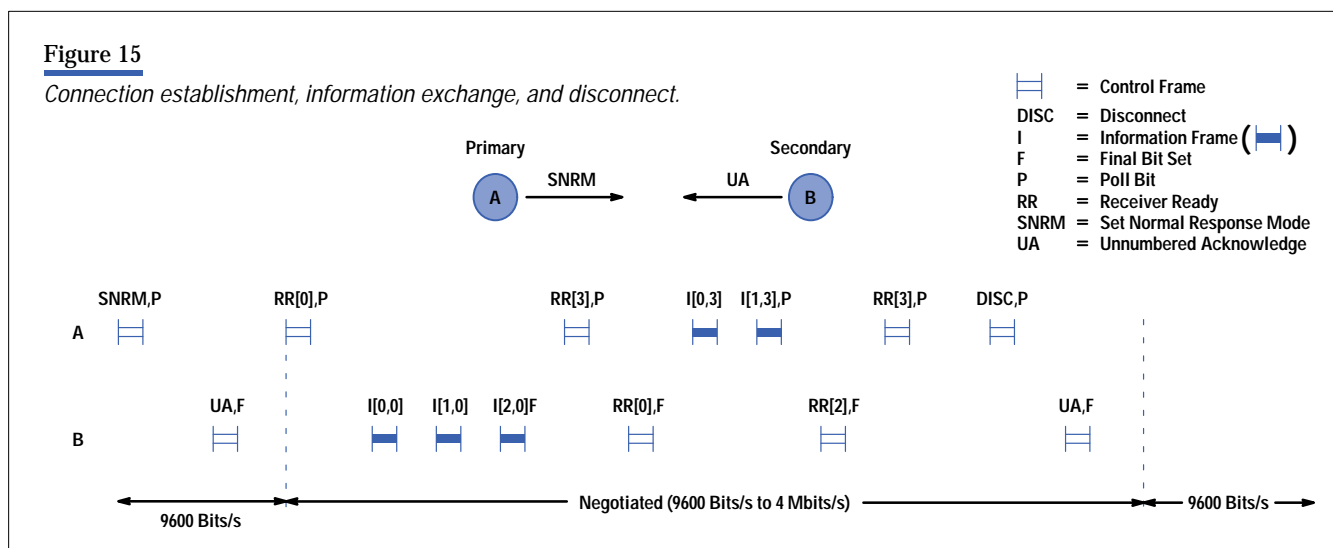


When the initial discovery XID command frame is received, devices B and C randomly choose slots in which to respond—in this example, slots 2 and 4. Device B then waits until it hears the discovery XID command indicating slot 2, and responds with a discovery XID response frame containing information about itself. Similarly, device C transmits a response during slot 4. Once the discovery process is over, all devices have the address and other information of all the devices within range: that is, device A has information about devices B and C, while devices B and C each have knowledge of device A. However, devices B and C are mutually hidden and as a result have no information about each other. This discovery information is passed to the upper layers whose responsibility it is to determine if there are any address collisions that need to be dealt with.

Should any of the devices that participated in the discovery process have duplicate addresses, then an address resolution process can be initiated. Address resolution follows a procedure similar to the discovery process, except that the device detecting the address conflict initiates the procedure, and resolution involves only the devices that have conflicting addresses. In this case, the initiating device transmits an address resolution XID command directed at the conflicting address. Devices with this address select another random address and a slot in which to respond. The initiator transmits the slot markers as before, and the previously conflicting devices respond in the appropriate slot. Once the process is over, each device should have a unique address. In the unlikely event that an address conflict still exists, the procedure can be repeated.

**Connection Establishment.** Once the discovery and address resolution processes are complete, the application layer may decide that it wishes to connect to one of the discovered devices. To connect, the application layer will issue a connection request which will ultimately result in the appropriate IrLAP service primitive being invoked. The IrLAP layer connects to the remote device by transmitting a set normal response mode (SNRM) command frame with the poll bit set. This command informs the remote device that the source wishes to initiate the connection and the poll bit indicates that a response is required. Assuming the remote device can accept the connection, it responds with an unnumbered acknowledge (UA) response frame with the final bit set. This indicates that the connection has been accepted. Under normal circumstances, the device that initiates the connection (transmits the set normal response mode) will become the

master, or primary, device, and the other device will become the slave, or secondary device. An example of connection establishment is shown in **Figure 15**. The notation used in the frames in **Figure 15** has the general form I(x,y) and RR(y), where x is the sequence number of the information frame and y is the sequence number of the next frame the source device expects to receive from the destination device.



The connection establishment takes place in normal disconnect mode (9600 bits/s), and once this is completed, the two devices will be in normal response mode. While in normal response mode, the devices can exchange data at any IrDA defined rate. However, not all IrDA devices will support all IrDA data rates or link parameters. It is therefore necessary for the devices to negotiate the parameters for normal response mode during connection setup. IrDA has defined several link parameters that can be negotiated:

- Data rate
- Maximum turnaround time
- Data size
- Window size
- Number of additional start of frame symbols (BOFs)
- Minimum turnaround time
- Link disconnect threshold time.

Data rate defines the data transfer rate during normal response mode (9600 bits/s to 4 Mbits/s), while maximum turnaround time defines the length of time either device may transmit before giving the other device a chance to transmit (50, 100, 250, or 500 ms). Data size determines the maximum length of the data field in an information frame (64 to 2048 bytes), and, in combination with the retransmission window size, which defines the number of outstanding frames that may be unacknowledged, allows devices with only limited resources to restrict the rate at which they will receive data. Number of additional BOFs and minimum turnaround time relate to physical layer restrictions, while link disconnect threshold time determines how long a device will wait without receiving a response from another device before assuming the link has failed and informing the upper layer that the link has disconnected.



Well-defined rules exist that ensure that after the set normal response mode-UA exchange has been completed, both devices will know the negotiated normal response mode parameters. Once both devices are in normal response mode, the primary device polls the secondary device by transmitting a receiver ready (RR) frame with the poll bit set, thereby initiating the information exchange phase.

Information Exchange and Link Reset. The information exchange procedure operates in a **master-slave mode in which the primary device controls the secondary device's access to the medium**. The primary device issues command frames to the secondary device which responds with response frames. To ensure that only one device can transmit frames at any one time, a **permission-to-transmit token is exchanged between the primary and secondary devices**. The primary device passes the permission-to-transmit token to the secondary by sending a command frame with the poll bit set. The secondary device returns the token by transmitting a response frame with the final bit set. The secondary device can only retain the token while it is transmitting data, and it must return it to the primary device if it has no data to transmit or if it reaches the maximum turnaround time. The primary device, however, within the limits imposed by the maximum turnaround time, can hold the token even if it has no data to transmit.

Although the physical layer has been designed to provide a low bit error rate channel, the dynamic nature of the infrared connection results in a possibility that frames may be lost in transit because of corruption by noise. To cope with this, the IrLAP protocol uses a **sequenced information exchange scheme with acknowledgments**. Should a frame be corrupted by noise, the CRC will highlight this error and the frame will be discarded. At the IrLAP layer, this error will be detected by virtue of the noncontiguous sequence numbers on the information frames. **The IrLAP protocol implements an automatic repeat request strategy** in the same manner as HDLC with options of using stop and wait, go back to N, and selective reject retransmission schemes.<sup>13</sup> This strategy allows the IrLAP layer to provide an error-free, reliable link to the IrLMP layer. An example of an error-free information exchange between two devices is shown in **Figure 15**.

Under exceptional circumstances, however, it may not be possible for the IrLAP entities in each device to recover from an error condition while maintaining the sequenced delivery of error-free information (I) frames. In this case, the IrLAP entity is allowed to reset the link. This reset involves discarding any undelivered information and reinitializing the sequence numbers and timers for the link. Although this may result in the loss of data, which the higher-level layer must deal with, it does allow the link to recover without the need for a total disconnection.

Connection Termination. Once the data exchange has taken place, the IrLAP link may be disconnected by either the primary or secondary devices. Should the primary wish to disconnect, it sends a disconnect command to the secondary device with the poll bit set. The secondary responds by returning an unnumbered acknowledge frame with the final bit set. Both devices will now be in normal disconnect mode, and the default normal disconnect mode parameters (9600 bits/s data rate) will apply. If the secondary wishes to disconnect, it transmits a request disconnect response with the final bit set when it is polled by the primary. The primary will then respond by transmitting a disconnect command, and both devices will be in normal disconnect mode. An example of a primary-initiated disconnection is shown in **Figure 15**. Once the two devices are in normal disconnect mode, the medium is free for any other device to initiate the discovery, address resolution, or connection procedures.

#### The Infrared Link Management Protocol

The Link Management Protocol (IrLMP) is layered on top of IrLAP, and has two main functions: application and service discovery and multiplexing of application level connections over the single IrLAP connections. The IrLMP layer allows individual service users (applications) to connect and exchange information with similar entities in the peer device, independent of any other service users that may be using the IrLAP connection. The IrLMP layer provides multiple independent channels to the IrLMP layer in the remote device. The IrLMP layer also provides a service with which applications can locally register themselves and some significant parameters in an information base. Services are also provided that enable those applications to access equivalent information in the information base of remote devices. Using this service, an application does not need prior knowledge of the applications in a remote device. This is an extremely useful feature for the kind of ad-hoc interactions typical of IrDA devices.



The two main functions provided by IrLMP are split between two sublayers. The Link Management Multiplexer (LM-MUX) provides the facilities for multiplexing application level connections over a an IrLAP connection between a pair of devices. The Link Management Information Access Service (LM-IAS) provides the services necessary to allow applications to discover devices and access the information in the information base of a remote device.

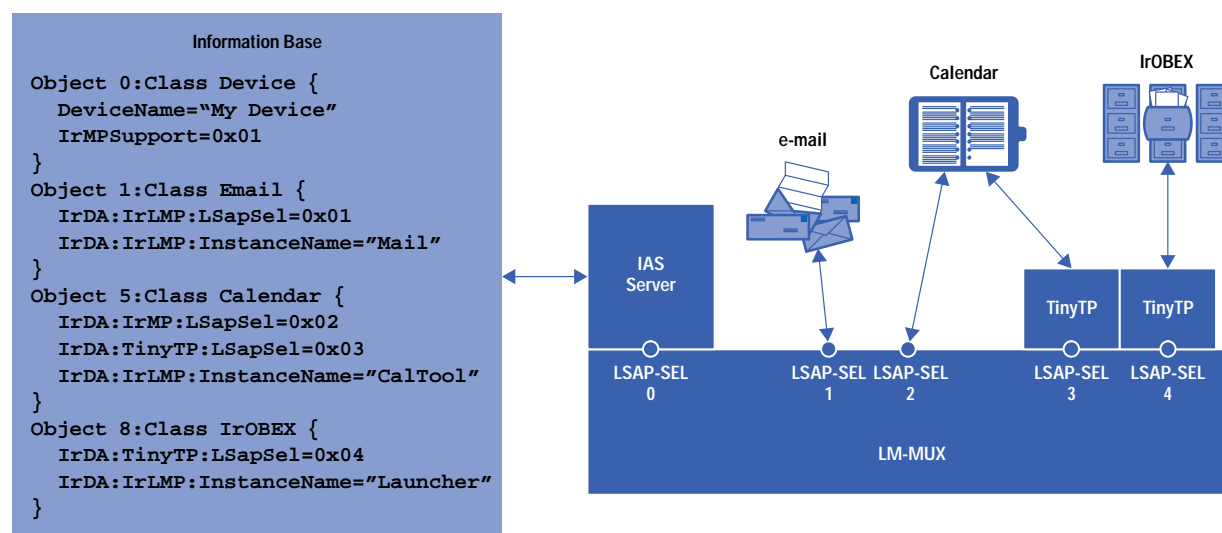
**The Link Management Multiplexer.** The LM-MUX adds two bytes of overhead to the IrLAP information frame, which are primarily used for addressing the individual multiplexed connections. The address fields uniquely identify the link service access points (LSAPs) in both the source and destination devices. Each LSAP is addressed by a seven-bit selector (LSAP-SEL), and LSAP-SELS within the range 0x01 to 0x6F can be used by applications. LSAP-SELS 0x00 and 0x70 are reserved for the information access service server and the connectionless data service respectively. The remaining LSAP-SEL values, 0x71 to 0x7F, are reserved for future use. Connections between IrLMP service users are called LSAP connections, and although an LSAP may terminate other LSAP connections, there is only one LSAP connection between any pair of LSAPs. All LSAP connections use the single IrLAP connection between the pair of devices.

**Information Access Service.** The information access service maintains information about the services provided by the host device and provides services that allow access to the information base on remote devices. The information access service allows devices to discover which services are available on the host device and provides the configuration information necessary to access those services. As an example, the most common piece of information required is the LSAP-SEL value, which tells where a particular service is located.

The information stored in the information base consists of a number of objects. Each object belongs to a specific class, and there may be several objects of the same class in the information base. The class defines the attributes that are present in each object, and these attributes can be assigned a particular value. The attributes of a class can be of type user string, octet sequence, signed integer, or missing. **Figure 16** shows an example of the information access service database for a device offering three unique services.

**Figure 16**

*Example information access service database.*



The example shows a device with three individual applications: e-mail, calendar, and IrOBEX (file transfer application). The information base contains three objects associated with these applications. The required Object 0 is always present within the information access service database, and it provides information about the device name and the version of IrLMP the device supports. All other devices can address Object 0 to get this information. Objects in the information base typically detail information about the services provided—for example, the LSAP-SEL where these services can be accessed. In the case of the calendar application, this service can be accessed using the Tiny TP flow-control mechanism on LSAP-SEL 3, or directly on LSAP-SEL 2. The difference is encoded in the attribute name.

The IrLMP layer provides several service primitives to access information access service data. However, the only mandatory service is GetValueByClass. This service requires the service user to provide the class and attribute names of the service it is interested in. The service returns a list of object identifiers and attribute values for all objects in the information base with the requested class and attribute name. Referring to the example in **Figure 16**, if a peer device issued a GetValueByClass with parameter Calendar for the class name and IrDA:IrLMP:LSapSel for the attribute name, the service would return a single element list with the entry containing object identifier 5 and attribute value 2.

#### Tiny TP Flow Control Mechanism

Although the IrLAP layer does have provisions for flow control, its use can result in deadlock situations, particularly where more than one IrLMP connection is operating. Such a deadlock situation can occur if an application in one device is waiting for its peer application to send it some data before releasing its buffer space. However, another connection may use up the remaining buffer space, causing the IrLAP layer to flow-control the link until buffer space becomes available. If both connections are waiting for data from the remote device before freeing the buffers, then clearly a deadlock has occurred that cannot be resolved without some form of higher-level intervention such as a system reset.

To overcome this problem, IrDA provides the lightweight transport protocol called Tiny TP.<sup>8</sup> Tiny TP adds a single byte of overhead to each frame and provides a per-LSAP-connection credit-based flow control mechanism with the possible segmentation and reassembly of service data units of up to 4 Gbytes in size. When a Tiny TP connection is initiated, the maximum service data unit size is negotiated and some initial credit is extended to each connection endpoint. Sending data causes the credit to be decreased by one, and periodically the receiver issues more credit. Without credit, the transmitter cannot send any data. It must wait until such times as the receiver extends it some more credit. Using Tiny TP, a device can ensure that credit is distributed among its applications, ensuring that the applications can communicate without reducing the buffer space to such a degree that IrLAP flow control must be used.

#### Conclusion

IrDA has completed the core standards necessary to enable any mobile computing platform with ad-hoc, point-and-shoot infrared communications from 2400 bits/s to 4 Mbits/s. Support for the IrDA platform from a wide variety of manufacturers is now becoming apparent, as many products—ranging from printers to laptop PCs and PDAs to mobile phones—are being released with IrDA capability. All these devices will have the ability to interoperate with one another should that be required. With over 130 companies actively maintaining membership in IrDA, currently released IrDA-enabled products represent only the tip of the iceberg. In the coming months and years, it is expected that more and more computing and other devices will be released with built-in IrDA capability.

However, providing the hardware platform to support IrDA is only half the story. Current activity within IrDA is directed at finishing off the IrDA series of standards to enable application-level developers to access the IrDA features in a uniform and efficient manner. The needs of legacy serial/parallel applications have been addressed with the IrCOMM standard. Legacy networking applications will be able to use the IrDA features implemented in the forthcoming IrLAN protocol. However, it is expected that a new class of applications will be developed with the express purpose of using the unique features of IrDA-enabled devices. The IrOBEX protocol, when completed, will provide application programmers with a generic method by which data can be exchanged with other applications without having to know the details of the destination application. As an example, transferring a graphic to another PDA (which will display it) or to a printer

(which will print it) will be no different from the source application's point of view. Alternately, a more flexible approach to accessing the IrDA communications facilities will be to directly access them through the operating system's application programming interface. An example of this is the WinSock-style API to IrDA, called IrSock,<sup>15</sup> currently being developed for the Microsoft® Windows 95 operating system.

In conclusion, the future for infrared is bright. With cross-industry support, IrDA is fast becoming the ubiquitous infrared communications system for portable and peripheral devices. Although legacy support for other infrared systems will persist for some time to come, the IrDA standard is now used on so many platforms that it is unlikely any new systems will be anything other than IrDA-enabled.

## References

1. *IrDA Marketing Requirements—Basis for the IrDA Technical Standards*, Version 3.2, The Infrared Data Association, November 23, 1993.
2. *Serial Infrared (SIR) Physical Layer Link Specification*, Version 1.0, The Infrared Data Association, April 27, 1994.
3. *Serial Infrared Link Access Protocol (IrLAP)*, Version 1.0, The Infrared Data Association, June 23, 1994.
4. *Link Management Protocol (IrLMP)*, Version 1.0, The Infrared Data Association, August 12, 1994.
5. *Serial Infrared Physical Layer Link Specification*, Version 1.1, The Infrared Data Association, October 17, 1995.
6. *Link Management Protocol (IrLMP)*, Version 1.1, The Infrared Data Association, January 23, 1996.
7. *Serial Infrared Link Access Protocol (IrLAP)*, Version 1.1, The Infrared Data Association, June 16, 1996.
8. *Tiny TP: A Flow-Control Mechanism for use with IrLMP*, Version 1.0, The Infrared Data Association, October 25, 1995.
9. *IrCOMM: Serial and Parallel Port Emulation over IR (Wire Replacement)*, Version 1.0, The Infrared Data Association, November 7, 1995.
10. *LAN Access Extensions for Link Management Protocol (IrLAN)*, Proposal, Version 1.0, The Infrared Data Association, January 1, 1996.
11. *Object Exchange Protocol (IrOBEX)*, Draft, Version 0.5f, The Infrared Data Association, July 24, 1996.
12. S. L. Harper and R. S. Worsley, "The HP 48SX Calculator Input/Output System, *Hewlett-Packard Journal*, Vol. 42, no. 3, June 1991.
13. *Information technology—Telecommunications and information exchange between systems—High-level data link control (HDLC) procedures—Elements of procedures*, ISO/IEC 4335, International Organization for Standardization, December 15, 1993.
14. *Information technology—Telecommunications and information exchange between systems—High-level data link control (HDLC) procedures—Classes of procedures*, ISO/IEC 7809, International Organization for Standardization, December 15, 1993.
15. *Infrared Sockets (IrSockets) Specification: Functional Specification*, Revision 1.0, The Microsoft Corporation, July 10, 1996.

Microsoft is a U.S. registered trademark of Microsoft Corporation.

**Iain Millar**

As a member of the technical staff at HP Laboratories in Bristol, England,

Iain Millar is involved in the development of protocols for the next generation of IrDA systems. He attended the University of Aberdeen in Scotland where he received a BSc (Eng.) degree (1988) in electrical and electronic engineering and a PhD (1995) in the area of fault tolerant protocols for LANs.

**Martin Beale**

Martin Beale is a member of the technical staff at Hewlett-Packard Labora-

tories in Bristol, England. He is working on the physical layer for the next generation IrDA systems. He earned a PhD degree (1994) in reduced complexity decoding of convolutional codes from the University of Cambridge. Outside of work he enjoys rock climbing, walking, cycling, skiing.

**Bryan J. Donoghue**

Bryan Donoghue is a member of the technical staff at HP Laboratories

in Palo Alto where he is working on the digital system design for high-speed wireless radio LANs. He received a MEng degree (1991) in electrical and electronic engineering from Loughborough University in England. Bryan was born in Llanelli, Wales and outside work he enjoys traveling, learning foreign languages, backpacking, and cycling.

**Kirk W. Lindstrom**

A member of the technical staff at HP's Communication Semiconductor

Solutions Division, Kirk Lindstrom is responsible for the design of ICs for infrared products. He joined HP in 1978 and since that time he has worked on the design of optocouplers, fiber-optic modules, and infrared transceivers. He holds a BS degree in EECS (1979) from the University of California at Berkeley. Besides being an ardent windsurfer, he also has an interest in doing and writing about investing.

**Stuart Williams**

A project manager at HP Laboratories in Bristol, England, Stuart Williams

is responsible for the infrared communications group. He has worked on various infrared protocol-related projects since he joined HP in 1992. He has a PhD degree (1986) from the University of Bath. Stuart was born in Rugby, Warwickshire, England, is married and has two sons. Biking and sailing occupy his free time.

- 
- ▶ [Go to Next Article](#)
  - ▶ [Go to Journal Home Page](#)