these types of networks do not observe all traffic that traverses the network; rather, the network itself contains some intelligence when it comes to delivering a packet. A node in a logically starred network sends a packet to a central hub that examines the destination address and then forwards the message to only the specified node. A mesh network routes traffic partly like a ring and partly like a star; however, multiple paths between nodes exist to complicate the delivery process. Based on the destination address, the originating node sends a packet to one of its neighbors, which in turn forwards the message to one of its neighbors. This process continues until the path has been completed and the packet arrives at its intended destination. The presence of multiple valid paths between nodes requires the mesh network to use knowledge about the location of nodes to select an optimal path through the network.

Access sharing is necessary on networks to ensure that each node eventually has an opportunity to send a message. Numerous methods of access sharing have been implemented over the years. Generally speaking, the length of messages is bounded to prevent a node from transmitting an infinitely long set of data and preventing anyone else from gaining access to the shared medium. Sharing algorithms differ according to the specific network topology involved. Networks that are a collection of point-to-point links (e.g., ring, star, mesh) do not have to worry about multiple nodes fighting for access to the same physical wire, but do have to ensure that one node does not steal all the bandwidth from others. Bus networks require sharing algorithms that address both simultaneous physical contention for the same shared wire in addition to logical contention for the network's bandwidth. Arbitration schemes can be centralized (whereby a single network master provides permission to each node to transmit) or distributed (whereby each node cooperates on a peer-to-peer level to resolve simultaneous access attempts).

After deciding on a network topology, one of the first issues to resolve is the network packet format. If the network type is already established (e.g., Ethernet), the associated formats and protocols are already defined by industry and government standards committees. If an application benefits from a simple, custom network, the packet format can be tailored to suit the application's specific needs.

Delineation and addressing are the two most basic issues to resolve. Delineation can be accomplished by sending fixed-size packets, embedding a length field in the packet header, or by reserving unique data values to act as start/stop markers. Framing with unique start/stop codes places a restriction on the type of data that a packet can contain: it cannot use these unique codes without causing false start or end indications. Referring back to Table 5.1, notice that start-of-header (SOH) and end-of-transmission (EOT) are represented by 0x01 and 0x04. These (or other pairs of codes) can be used as delimiters if the packet is guaranteed to contain only alphanumeric ASCII values that do not conflict with these codes.

Addressing is normally achieved by inserting both the destination and source addresses into the header. However, some networking schemes may send only a single address. Sending both addresses enables recognition of the destination as well as a determination of which node sent the packet. Since most data exchanges are bidirectional to a certain degree, a destination node will probably need to send some form of reply to the source node of a particular packet. Many networks include a provision known as *broadcast addressing* whereby a packet is sent to all nodes on the network rather than just one. This broadcast is often indicated using a reserved *broadcast* address. In contrast to a *unicast* address that is matched by only one node, a broadcast address is matched by all nodes on the network. Some networks also have *multicast* addresses that associate multiple nodes with a single destination address.

5.9 RS-485

Whereas RS-232 and RS-422 enable point-to-point serial links, the RS-485 standard enables multiple-node networks. Like RS-422, RS-485 provides differential signaling to enable communications across spans of twisted-pair wire exceeding 1.2 km. Unlike RS-422, the RS-485 standard allows up to 32 transmit/receive nodes on a single twisted pair that is terminated at each end as shown in Fig. 5.14. Modern low-load receivers that draw very little current from the RS-485 bus can be used to increase the number of nodes on an RS-485 network well beyond the original 32-node limit to 256 nodes or more. A single pair of wires is used for both transmit and receive, meaning that the system is capable of *half-duplex* (one-way) operation rather than *full-duplex* operation (both directions at the same time). Half-duplex operation restricts the network to one-way exchange of information at any given time. When node A is sending a packet to node B, node B cannot simultaneously send a packet to node A.

RS-485 directly supports the implementation of bus networks. Bus topologies are easy to work with, because nodes can directly communicate with each other without having to pass through other nodes or semi-intelligent hubs. However, a bus network requires provisions for sharing access to be built into the network protocol. In a centralized arbitration scheme, a master node gives permission for any other node to transmit data. This permission can be a request-reply scheme whereby slave nodes do not respond unless a request for data is issued. Alternatively, slave nodes can be periodically queried by the master for transmit requests, and the master can grant permissions on an individual-node basis. There are many centralized arbitration schemes that have been worked out over the years.

A common distributed arbitration scheme on a bus network is *collision detection* with random back-off. When a node wants to transmit data, it first waits until the bus becomes idle. Once idle, the node begins transmitting data. However, when the node begins transmitting, there is a chance that one or more nodes have been waiting for an opportunity to begin transmitting and that they will begin transmitting at the same time. Collision detection circuits at each node determine that more than one node is transmitting, and this causes all active transmitters to stop. Figure 5.15 shows the imple-

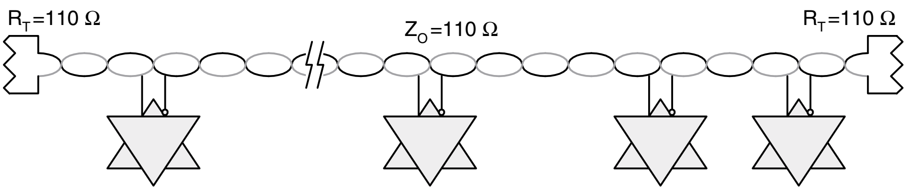

FIGURE 5.14 RS-485 bus topology.

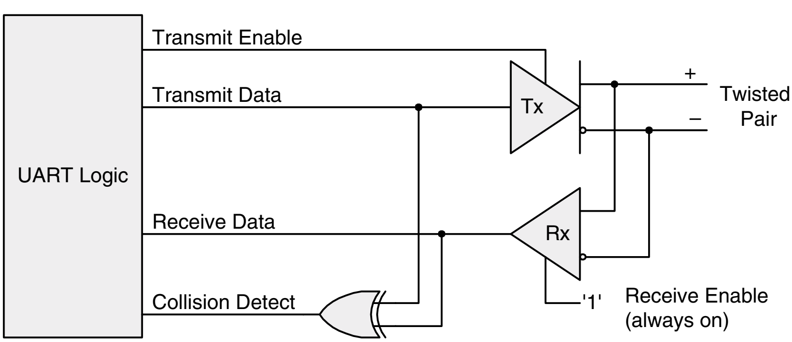

FIGURE 5.15 RS-485 collision detection transceiver.