mentation of an RS-485 transceiver with external collision detection logic. A transmit enable signal exists to turn off the transmitter when the UART is not actively sending data. Unlike an RS-422 transmitter that does not have to share access with others, the RS-485 transmitter must turn itself off when not sending data to enable others to transmit.

When transmitting, the receiver returns the logical state of the twisted-pair bus. If the bus is not at the same state as the transmitted data, a collision is most likely being caused by another transmitter trying to drive the opposite logic state. An XOR gate implements this collection detect, and the XOR output must be sampled only after allowing adequate time for the bus to settle to a stable state following the assertion of each bit from the transmitter.

Once a collision has been detected by each node and the transmitters are disabled, each node waits a different length of time before retransmitting. If all delays were equal, multiple nodes would get caught in a deadlock situation wherein each node keeps trying to transmit after the same delay interval. Random back-off delays are pseudo-random so as to not unfairly burden some nodes with consistently longer delays than other nodes. At the end of the delay, one of the nodes begins transmitting first and gains control of the bus by default. The other waiting nodes eventually exit from their delays and observe that the bus is already busy, indicating that they must wait their turn until the current packet has been completed. If, by coincidence, another node begins transmitting at the same time that the first node begins, the back-off process begins again. It is statistically possible for this process to occur several times in a row, although the probability of this being a frequent event is small in a properly designed network. A bus network constructed with too many nodes trying to send too much data at the same time can exhibit very poor performance, because it would be quite prone to collisions. In such a case, the solution may be to either reduce the network traffic or increase the network's bandwidth.

5.10 A SIMPLE RS-485 NETWORK

An example of a simple but effective network implemented with RS-485 serves as a vehicle to discuss how packet formats, protocols, and hardware converge to yield a useful communications medium. The motivation to create a custom RS-485 network often arises from a need to deploy remote actuators and data-acquisition modules in a factory or campus setting. A central computer may be located in a factory office, and it may need to periodically gather process information (e.g., temperature, pressure, fluid-flow rate) from a group of machines. Alternatively, a security control console located in one building may need to send security camera positioning commands to locations throughout the campus. Such applications may involve a collection of fairly simple and inexpensive microprocessor-based modules that contain RS-485 transceivers. Depending on the exact physical layout, it may or may not be practical to wire all remote nodes together in a single twisted-pair bus. If not, a logical bus can be formed by creating a hybrid star/bus topology as shown in Fig. 5.16. A central hub electrically connects the individual star segments so that they function electrically as a large bus but do not require a single wire to be run throughout the entire campus.

As shown, the hub does not contain any intelligent components—it is a glorified junction box. This setup is adequate if the total length of all star segments does not exceed 1.2 km, which is within the electrical limitations of the RS-485 standard. While simple, this setup suffers from a lack of fault tolerance. If one segment of the star wiring is damaged, the entire network may cease operation because, electrically, it is a single long pair of wires. Both the distance and fault-tolerance limitations can be overcome by implementing an active hub that contains *repeaters* on each star segment and smart switching logic to detect and isolate a broken segment. A repeater is an active two-port device that amplifies or regenerates the data received on one port and transmits it on the other port. An RS-

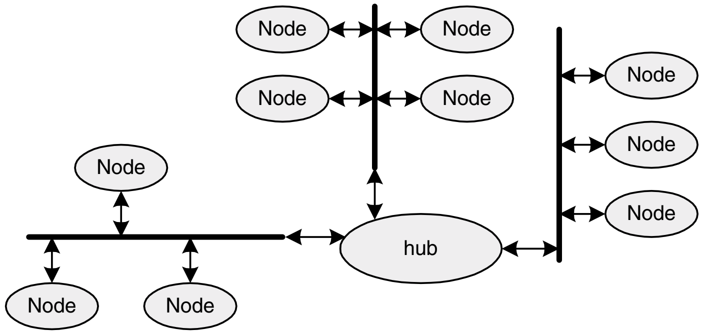

FIGURE 5.16 Hybrid star/bus network topology.

485 repeater needs a degree of intelligence, because both ports must be bidirectional. Therefore, the repeater must be able to listen for traffic on both sides, detect traffic on one side, and then transmit that traffic on the other side. A hub that detects and isolates segment failures would be well designed to report this fault information to a central control node to alert the human operator that repairs are necessary. These possible improvements in the network hub do not affect the logical operation of the network and, consequently, are not a focus of this discussion.

With a topology chosen and a general application in mind, the next step is to decide on the network's operational requirements from among the following:

1. *Support for roughly 200 nodes* provides flexibility for a variety of control applications.
2. *Central arbitration handled by master control node* for simplicity of network design. A facility control network is often a master-slave application, because all data transfers are at the request of the central controller. Central arbitration removes the need for collision-detect hardware and random back-off algorithms.
3. *Broadcast capability* enables easy distribution of network status information from the master control node.
4. *Data rate of 9600 bps* provides adequate bandwidth for small control messages without burdening the network with high frequencies that can lead to excessive noise and signal degradation.
5. *Basic error handling* prevents processing incorrect data and network lock-up conditions when occasional noise on the RS-485 twisted-pairs causes data bits to change state.

Many aspects of network functionality are directly influenced by a suitable network packet format. Other aspects are addressed by the protocol that formats data on the network, by the transceiver and UART hardware, or by a combination of these three elements.

In considering the packet format, 8-bit destination and source addresses are chosen to support more than 200 nodes on the network. A special destination address value of 0xFF represents a broadcast address, meaning that all nodes should accept the packet automatically. Such broadcast packets are useful for system-wide initialization whereby, for example, the control computer can send the current time to all nodes. This multicast address cannot be used as a normal node address, thereby limiting the network to 255 unique nodes.

It is desirable to employ variable-length packets so that a message does not have to be longer than necessary, thereby conserving network bandwidth. Variable-length packets require some mechanism