

Module 6 Metropolitan Area Networks

Lesson

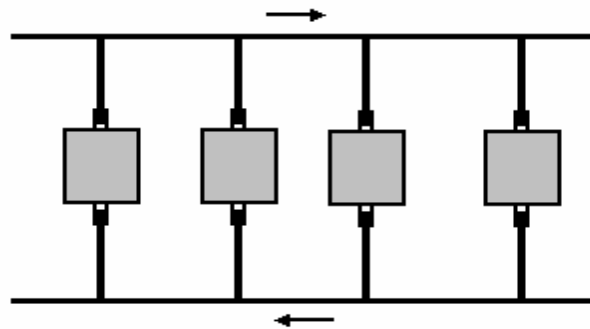
19

Distributed Queue Dual Bus (DQDB)

6.1.1 INTRODUCTION

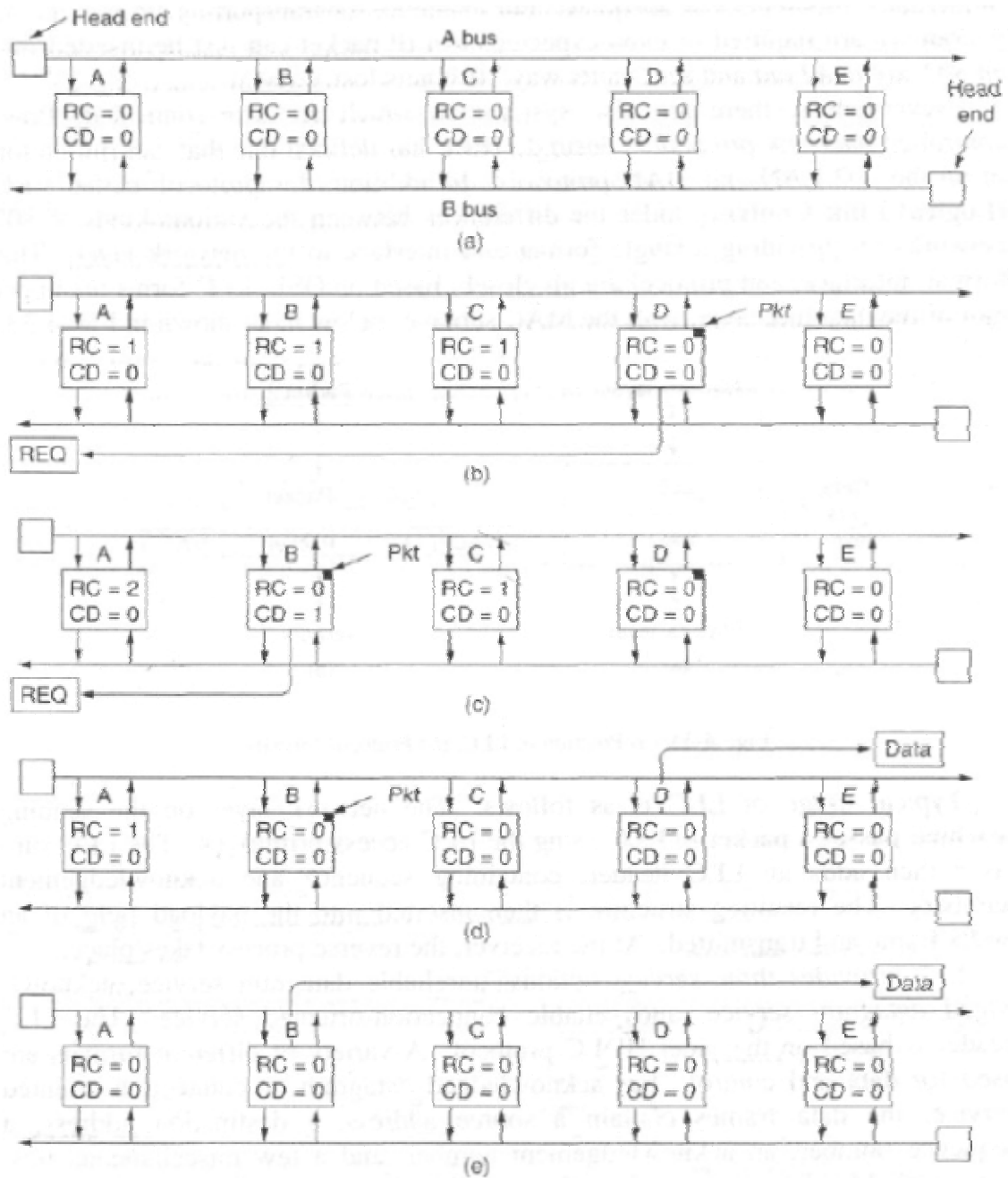
IEEE defined a MAN standard for covering an entire city. This was called **Distributed Queue Data Interface (DQDB)** and put up as IEEE standard 802.6.

The basic geometry of DQDB is as shown in the figure below. Two parallel unidirectional buses are laid down in the area to be covered by the network. The stations are attached to both the buses in parallel. Each bus has a head-end, which generates a steady stream of 53byte cells. Each cell travels downstream from the head end. When it reaches the end, it falls off the bus. Each cell carries a 44 byte payload field, making it compatible with some AAL modes. Each cell also holds two protocol bits, *Busy*, set to indicate that a cell is occupied, and *Request*, which can be set when a station wants to make a request.



To transmit a cell, a station has to know whether the destination is to the left of it or to its right. If the destination is to the right, the sender uses bus A. Otherwise it uses bus B. Data are inserted onto either bus using a wired-OR-circuit, so failure of a station does not take down the network.

The 802.6 protocol does not have greedy stations. In all others, if a station gets a chance to send, it sends. Here the queue up and become ready to transmit in FIFO order, all this without having a central queue. To simulate the FIFO queue, each station maintains two counters, *RC* and *CD*. *RC* (*Request Counter*) counts the number of downstream request pending until the station itself has a frame to send. At that point, *RC* is copied to *CD*, *RC* is reset to 0 and now counts the number of requests made after the station became ready. For example, if $CD = 3$ and $RC = 2$ for station k , the next three empty cells that pass by station k are reserved for downstream stations, then station k may send, then two more cells are reserved for downstream stations. For simplicity, we assume a station can have only one cell ready for transmission at a time.



To send a cell, a station must first make a reservation by setting the *Request* bit in some cell on the reverse bus (i.e., on bus *B* for a transmission that will later take place on bus *A*). As this cell propagates down the reverse bus, every station along the way notes it and increments its *RC*. To illustrate this concept, we will use an example. Initially, all the *RC* counters are 0, and no cells are queued up, as shown in Fig. 4-32(a). Then station *D* makes a request, which causes stations *C*, *B*, and *A*, to increment their *RC* counters, as shown in Fig. 4-32(b). After that, *B* makes

a request, copying its current RC value into CD , leading to the situation of Fig.4-32(c). At this point, the head-end on bus A generates an empty cell. As it passes by B , that station sees that its $CD > 0$, so it may not use the empty cell. (When a station has a cell queued, CD represents its position in the queue, with 0 being the front of the queue). Instead it decrements CD . When the still-empty cell gets to D , that station sees that $CD = 0$, meaning that no one is ahead of it on the queue, so it ORs its data into the cell and sets the *Busy* bit. After the transmissions are done, we have the situation of Fig. 4-32(d).

When the next empty cell is generated, station B sees that it is now at the head of the queue, and seizes the cell (by setting I bit), as illustrated in Fig. 4- 32(e). In this way, stations queue up to take turns, without a centralized queue manager.

Objective Questions

19.01 Each DQDB cell carries a _____ byte payload to make it compliant with _____ modes.

19.02 The 802.6 protocol is a greedy protocol – True/False

Subjective Questions

19.11 Explain the basic geometry of a DQDB system and its working.

19.12 Explain how data transmission occurs in an 802.6 LAN.

19.12 Explain the DQDB cell format.

Level 2 Questions

19.21