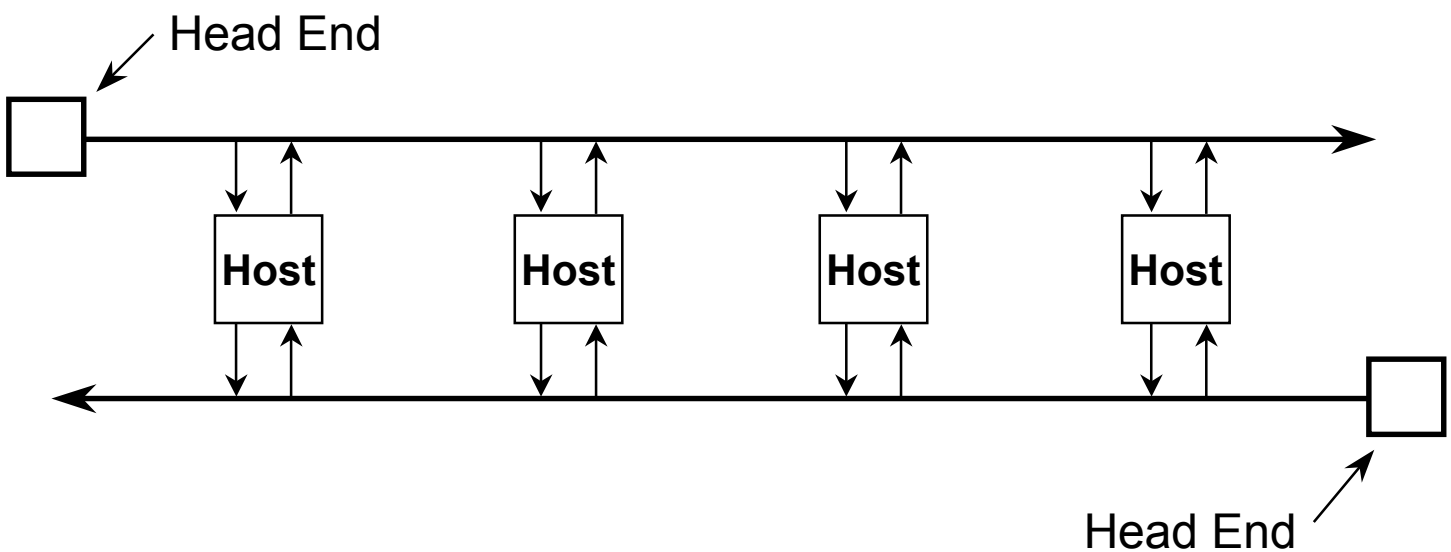
DQDB

- Distributed Queue Dual Bus (DQDB)
- DQDB is a MAN.
- Unlike FDDI, DQDB is an IEEE standard: 802.6

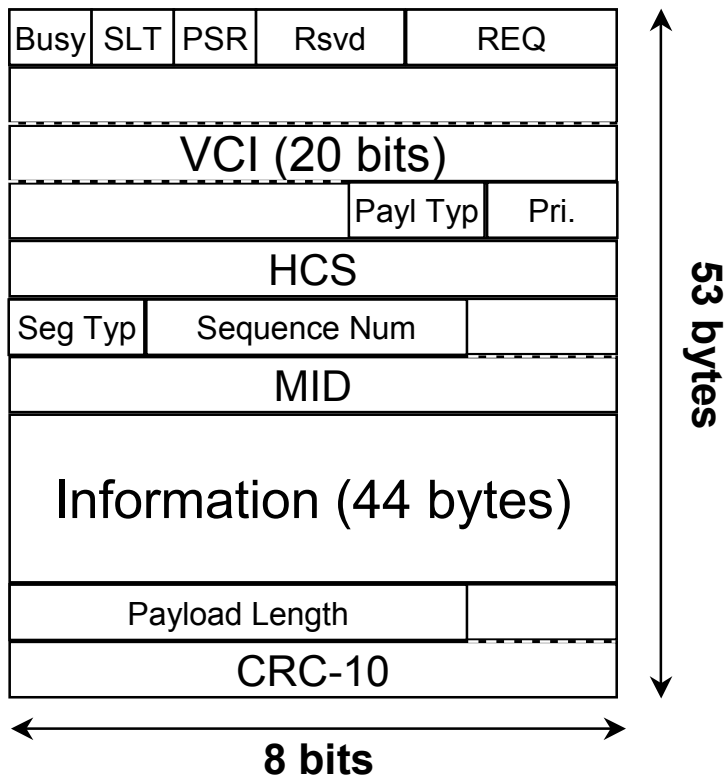
DQDB *(cont'd)*

- Topology: Dual Bus



DQDB Frame Format

53-byte frame called a “cell”

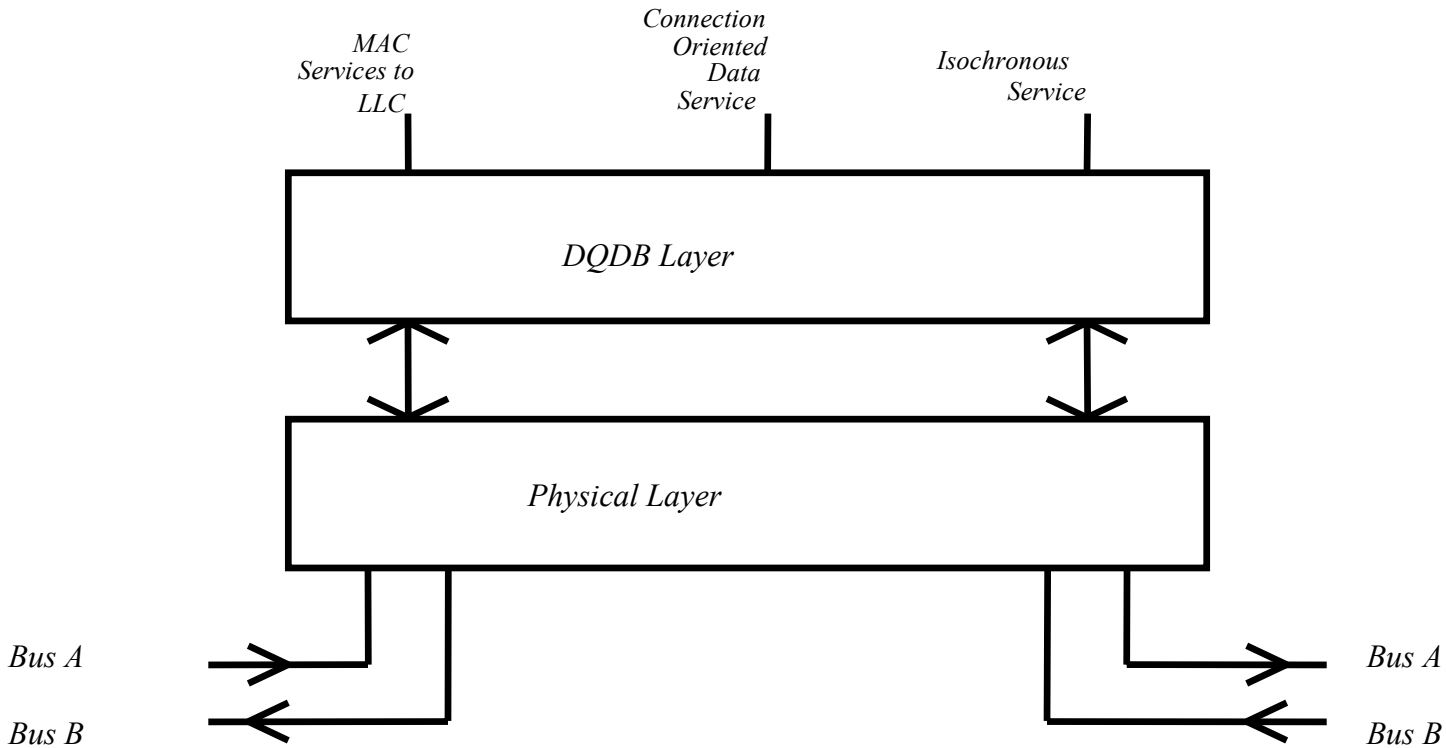


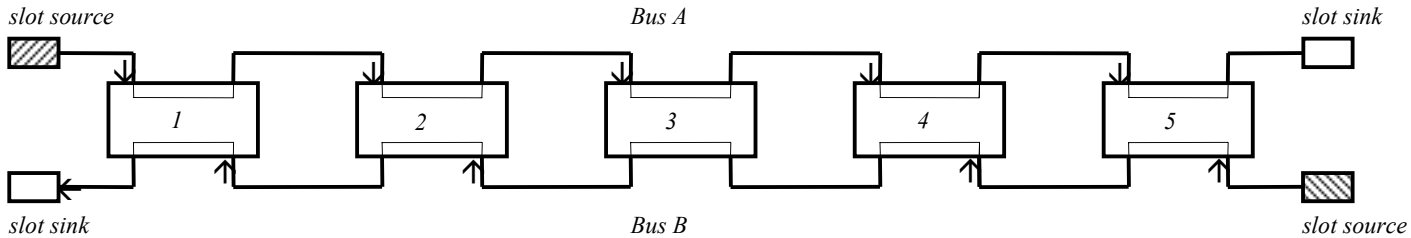
DQDB MAC Sublayer

- Head-ends generate cells in both directions
- To transmit, a host must know whether the destination is to its right or its left
  - If right, the host must send on one bus
  - If left, the host must send on the other bus
- A “Distributed Queue” is used to make sure that cells are transmitted on a first-come first-serve basis

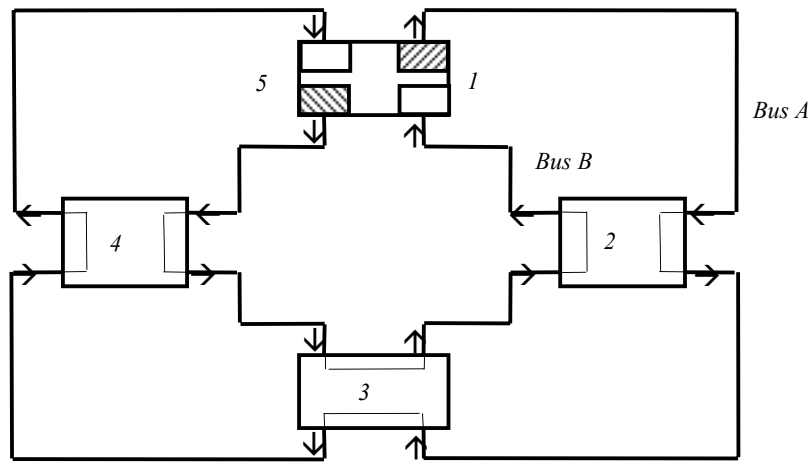
Distributed Queueing

- Each host keeps two counters, CD and RC
- Procedure for sending a cell:
  - Set the request bit in a non-busy cell passing on the reverse bus. Copy the RC counter to the CD counter, and reset the RC counter to zero.
  - As the request cell passes on the reverse bus, each host on the path observes it and increments its RC counter
  - When an empty cell passes on the forward bus, the hosts decrement their RC counters and CD counters
  - If RC and CD equal zero when an empty cell passes on the forward bus, it may be used to send the message


MAC
Services to
LLC
Connection
Oriented
Data
Service
Isochronous
Service
DQDB Layer
Physical Layer
Bus A
Bus B
Bus A
Bus B


slot source
Bus A
slot sink
1
2
3
4
5
slot sink
Bus B
slot source

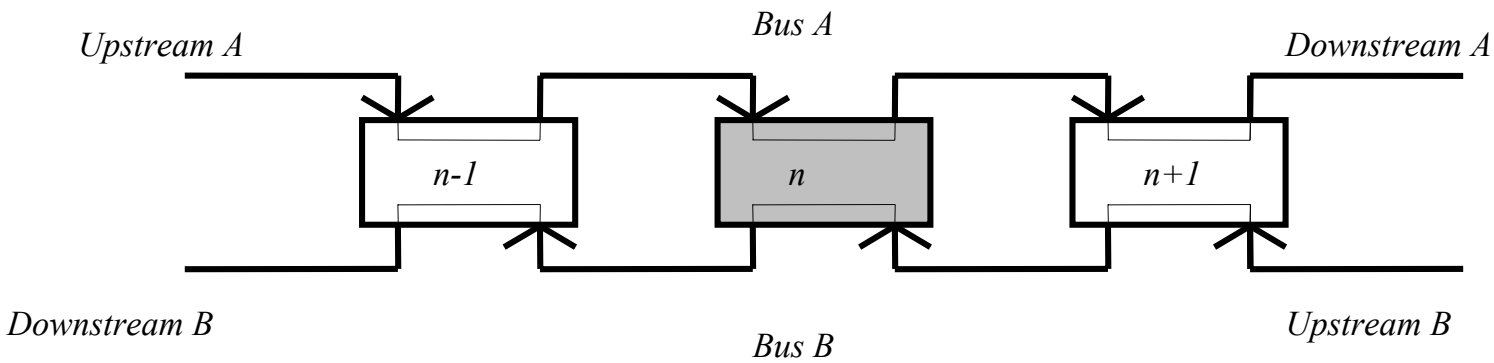
(a)


5
1
Bus A
Bus B
4
2
3

(b)

DQDB

- Transmitting Data
  - –Node acquires slot
  - –Sets header
  - –Copies data into slot
  - –Cells propagate to end of bus
    - (absorbed by sink)
  - –Copied by intended destination on way



The terms upstream and downstream are relative to node n

A fragment of a DQDB network

DQDB

- Services
  - –CONS services
    - Pre-Arbitrated (PA) (reserved)
  - –MAC layer services
    - Queue Arbitrated (QA) (available)
    - Distributed Queue
    - Node gains access when its slot reaches head of Q

  essentially distributed Q allows access to bus in an orderly way

DQDB

Distributed Queue

- Each node is aware of relative position of all other nodes
- Correct bus must be chosen to transmit data
- Distributed Q is independent of physical size of network

In general

–one bus to send data to nodes located to left

–other bus to send data to nodes located on right

–nodes competing with each other for access to bus

DQDB

Implementation of distributed Q

- 1st octet - Access Control Field
- Slot overhead - 2 bits (ACF)
  - –Busy (B) bit
    - set when node acquires slot
  - –Request (R) bit
    - set by node wishing to transmit

DQDB

- Node wishing to transmit on Bus A
  - Notify all nodes upstream that they are joining Q
    - Set first available R bit on Bus B
    - Why must nodes only notify upstream nodes?
    - How does node know when an available slot has its name on it?

DQDB

- When to transmit?

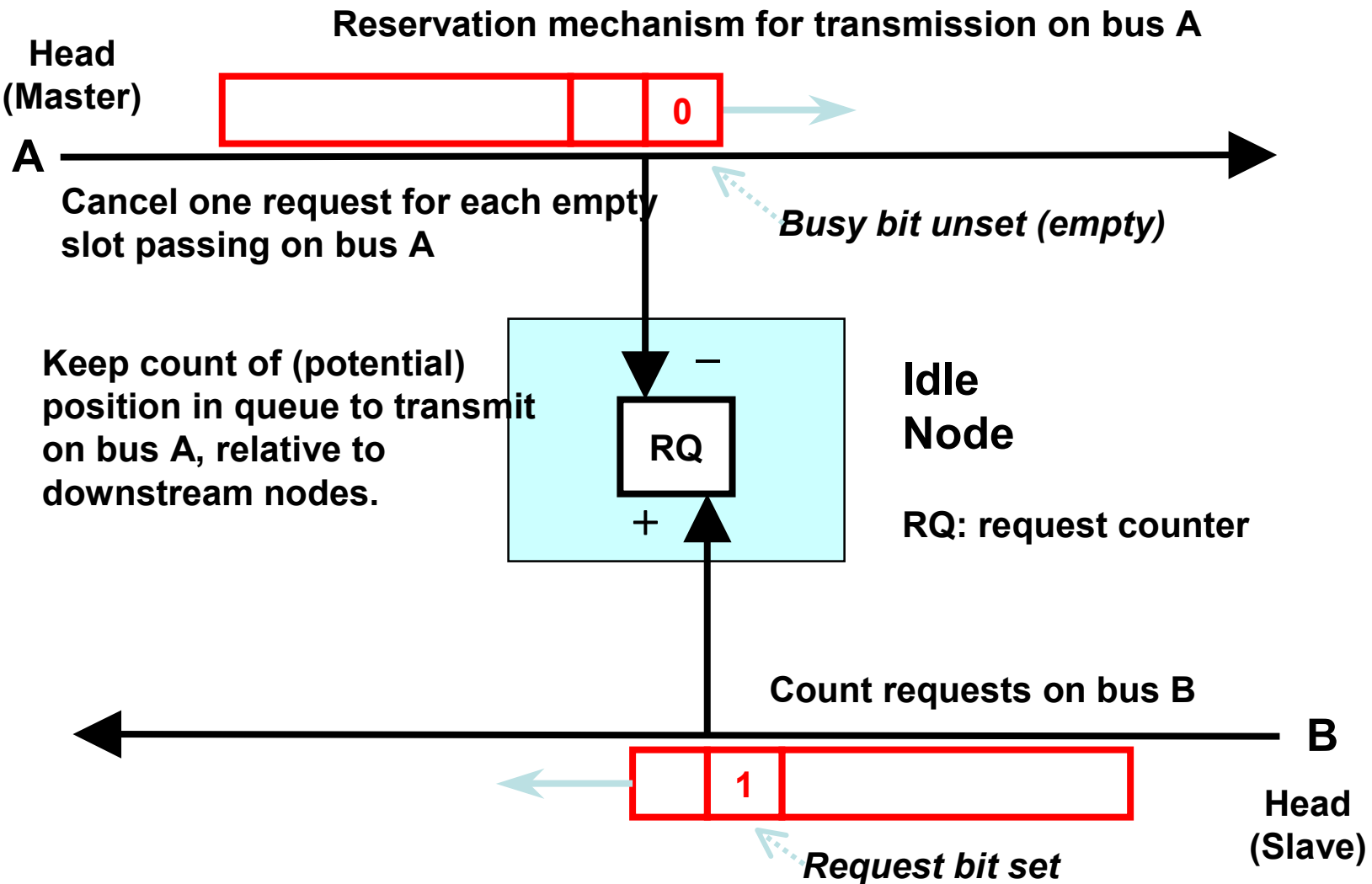
- When node is inactive
  - –counts requests passing on Bus B
  - –decrements count for every empty slot that passes on Bus A

  - –Thus, state of R counter at node shows length of Q of cells waiting to be transmitted by all downstream nodes

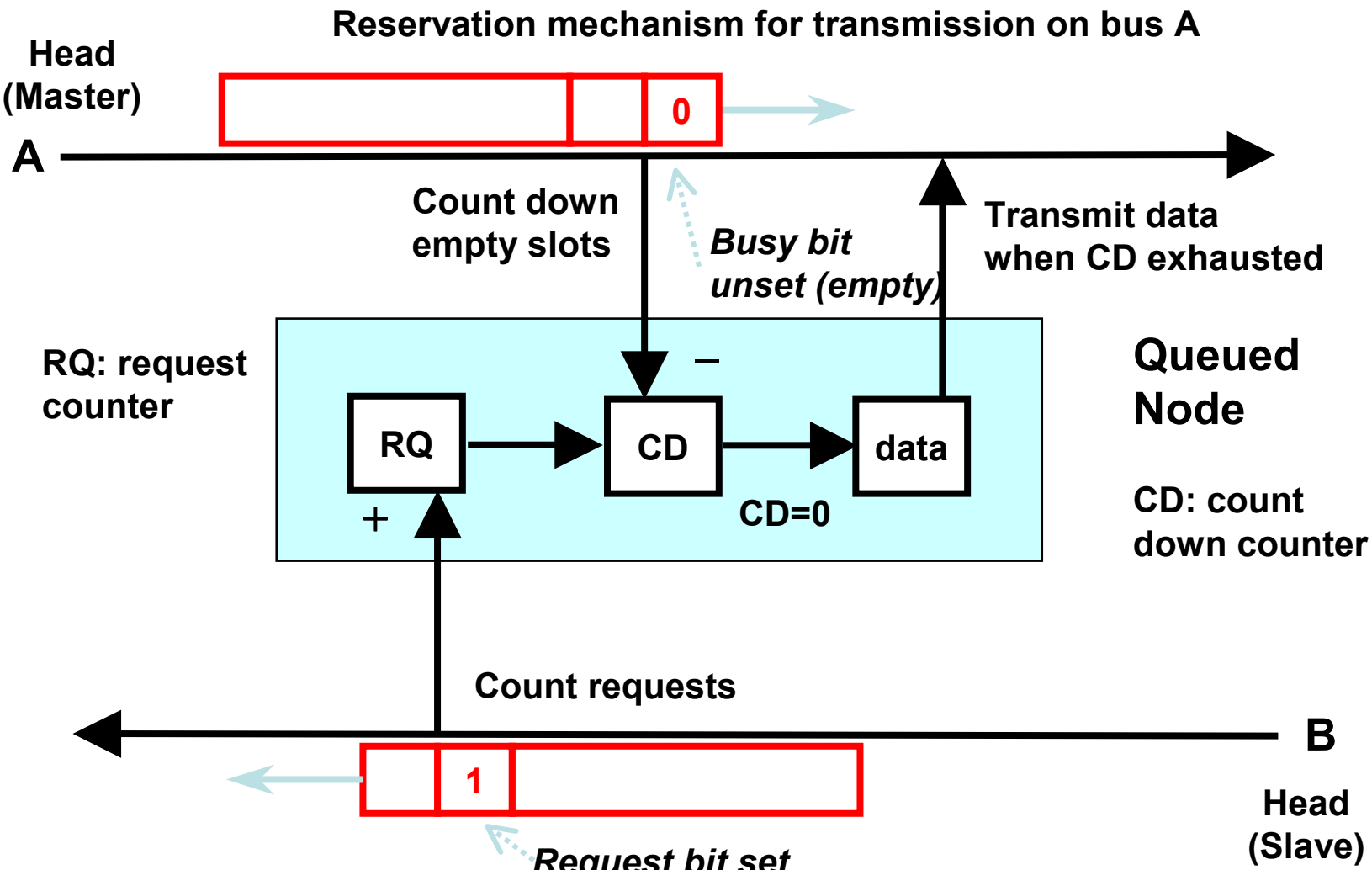
DQDB

- When node is ready to transmit
  - –Sets first available R bit on Bus B
    - (joining Q for Bus A)
  - –R Counter is copied to a separate Count-Down (CD) counter
  - –CD counter is decremented by every passing available slot passing on Bus A
  - –R counter is cleared and counting begins again.
    - (Gives length of Q behinf cell to be transmitted)
  - –CD counter of zero indicates the node has reached top of Q

DQDB: Distributed Queue



DQDB: Distributed Queue…(contd.)



DQDB: slots, data segments

- Slot types:
  - *Queued:* conventional data traffic (aysnchronous)
  - *Pre-arbitrated:* for isochronous support
    - 125μs frames, 8 kHz framing clock, allows TDM use:
      1 byte/frame equivalent to 64 kbps
      1 slot/frame equivalent to approx. 3 Mbps
- Packet or MAC frame segmentation
  - like ATM AAL3/4
- VCI
  - enables nodes to recognize which traffic to receive
  - in combination with byte offset in segment, identifies pre-arbitrated channel
  - channel slots allocated and preformatted by head ends
  - broadcast / multicast supported

DQDB: fairness, SMDS

- Queued access: fair for all stations?
  - No: stations at the ends receive different service to those in the middle; downstream stations may be starved by upstream activity.
  - Partial remedy:
    - slot re-use: enable use of slot by nodes downstream of receiver
    - complicates nodes
    - increases utilization
    - not a complete remedy
- DQDB provided basis for definition of SMDS service
  - SMDS carefully defined (by Bellcore) to be entirely compatible with subset of DQDB operation *and* technology independent