บทที่ 2

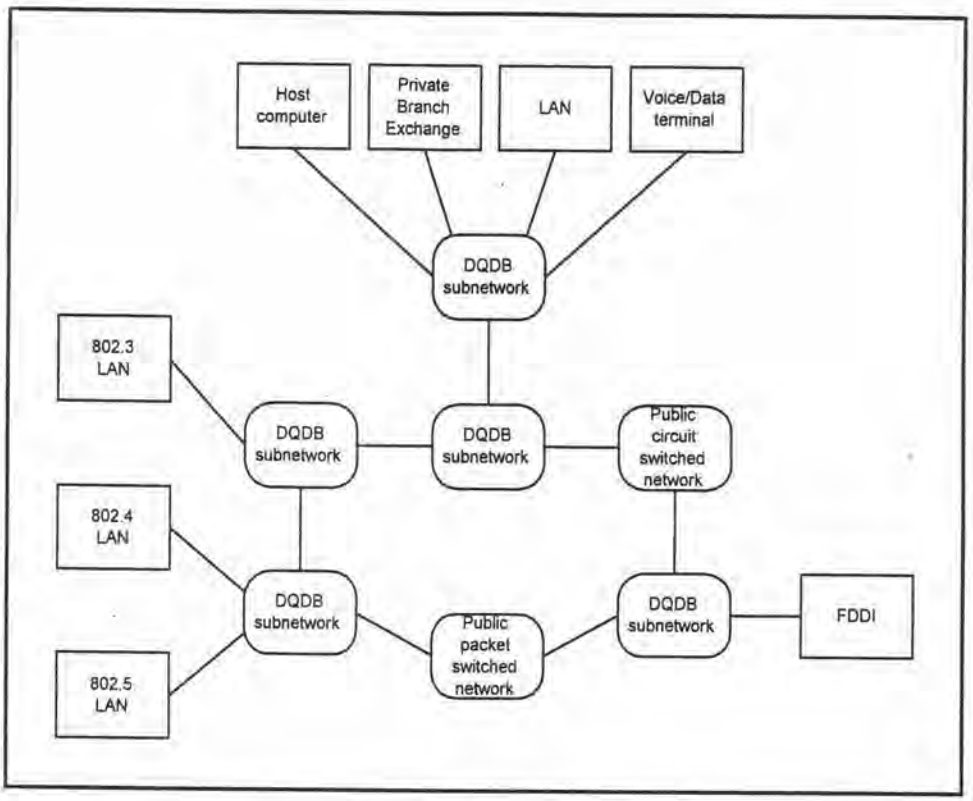
โปรโตคอล DQDB

2.1 บทนำ

มาตรฐาน DQDB ได้ถูกพัฒนาขึ้นโดยคณะอนุกรรมการ IEEE 802.6 โดยมีจุดประสงค์เพื่อให้ DQDB เป็นโครงข่ายของโครงข่าย MAN ที่สามารถให้บริการร่วม เช่น เสียง ข้อมูล และสัญญาณวิดีโอได้ โดยมีขอบเขตพื้นที่ในการให้บริการที่กว้างมากกว่า 50 กิโลเมตร ซึ่งโครงข่าย DQDB นี้สามารถนำมาเชื่อมต่อกันโดยใช้บริดจ์, เราเตอร์, เกตเวย์ หรือโครงข่ายอื่น ๆ เพื่อทำหน้าที่เป็นโครงข่าย MAN ได้ ดังแสดงในรูปที่ 2.1

คุณลักษณะที่สำคัญบางประการของ DQDB

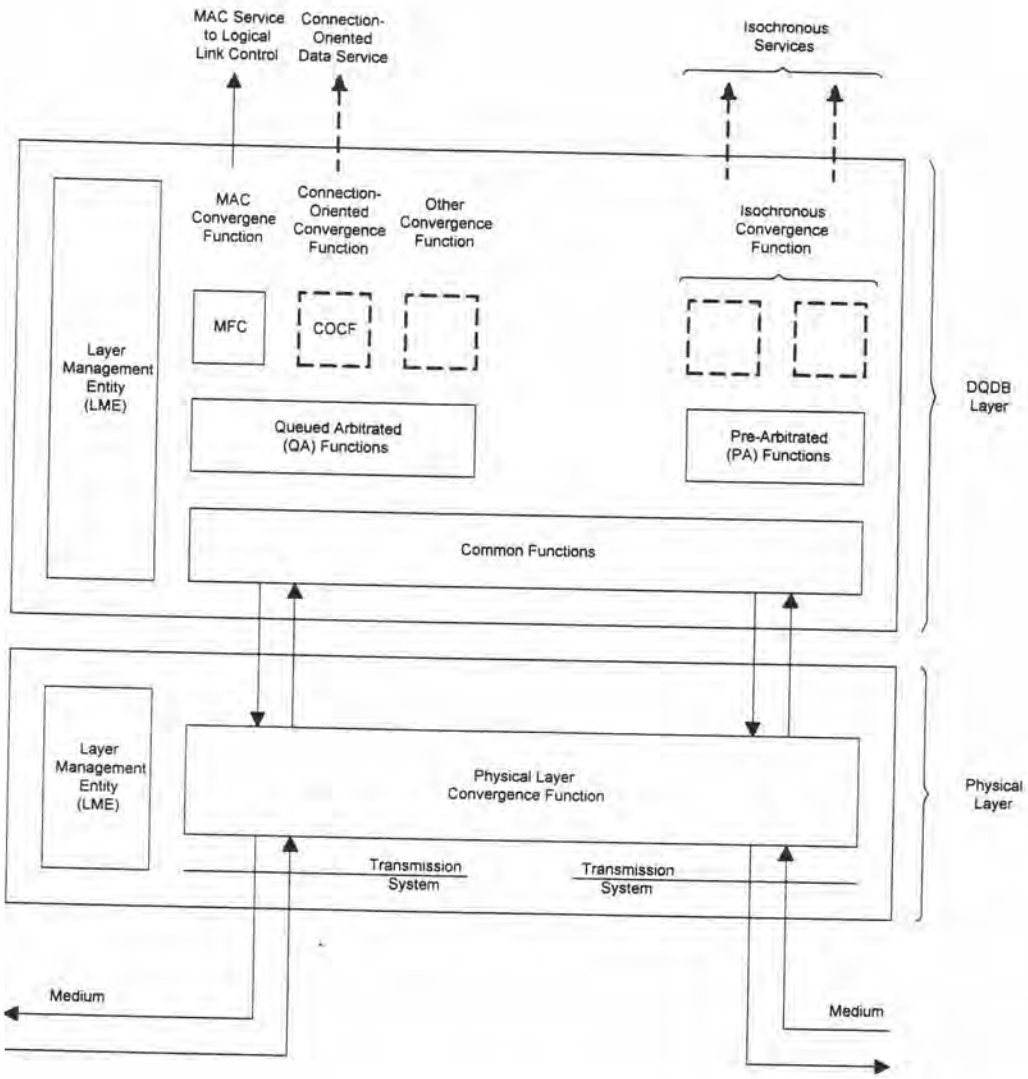
1. ใช้สถาปัตยกรรมแบบบัสคู่ ซึ่งการทำงานของแต่ละบัสจะเป็นอิสระต่อกัน
2. มีความเข้ากันได้กับ IEEE 802 LAN เพราะ DQDB สนับสนุน 802.2 LLC
3. สามารถใช้ตัวกลางได้หลายประการ เช่น สายเคเบิลแกนร่วม ไมโครเวฟ และสายเคเบิลเส้นใยแสง
4. มีความทนต่อความผิดพร่อง (Fault tolerance) เพราะมีการใช้โทโปโลยีแบบบัสคู่แบบวง (looped dual bus topology)
5. มีอัตราข้อมูลสูง คือ ตั้งแต่ 34 ถึง 155 Mbps และมากกว่า
6. มีความเข้ากันได้กับ ATM เพราะใช้แพ็กเกตข้อมูลที่เรียกว่า "สลอต (slot)" ซึ่งมีขนาด 53 ไบต์ เช่นเดียวกับ ATM
7. สามารถให้บริการแบบสวิตช์วงจร (circuit swiched service) และบริการแบบสวิตช์กลุ่มข้อมูล (packet swiched service) ได้



รูปที่ 2.1 โครงข่าย DQDB MAN

2.2 สถาปัตยกรรมของโปรโตคอล DQDB

มาตรฐาน IEEE 802.6 ได้กำหนดโครงข่ายความเร็วสูงที่มีการเข้าถึง (Access) ช่องสัญญาณแบบ DQDB ซึ่งมีความเข้ากันได้กับสถาปัตยกรรมของ IEEE 802 LAN เพราะต่างก็ทำงานภายใต้โปรโตคอล IEEE 802.2 LLC (Logical Link Control) และในส่วนของมาตรฐาน IEEE 802.6 ได้มีการกำหนดโปรโตคอลไว้ 2 ระดับ คือ ระดับชั้นทางกายภาพ (Physical layer) และระดับชั้น DQDB (DQDB layer) ดังรูปที่ 2.2



รูปที่ 2.2 สถาปัตยกรรมของโปรโตคอล DQDB

1. ระดับชั้นทางกายภาพ (Physical layer)

จะสอดคล้องตามระดับชั้นทางกายภาพของ OSI ซึ่งกำหนดวิธีการที่จะใช้ตัวกลางที่ความเร็วต่าง ๆ กัน ซึ่งมาตรฐานได้สนับสนุนระบบการส่ง (transmission

system) หลายประเภท เช่น DS-3 (44.736 Mbps), CEPT ได้ระดับ 3 และ 4 (34.368 และ 139.264 Mbps ตามลำดับ) และ SONET/SDH (155.52 Mbps และ มากกว่า) และในระดับชั้นนี้จะมี Physical layer convergence protocol (PLCP) ซึ่งจะทำหน้าที่เป็นตัวปรับความสามารถในการให้บริการของระบบการส่ง เพื่อให้ระบบการส่งนั้นๆ สามารถให้บริการแก่ระดับชั้น DQDB ได้ โดย PLCP จะทำการแมป (map) รูปแบบของโปรโตคอล DQDB ให้เหมาะสมกับรูปแบบของแต่ละระบบการส่ง ดังนั้นแต่ละระบบการส่งก็จะต้องมี PLCP ที่แตกต่างกัน

2. ระดับชั้น DQDB (DQDB layer)

จะเป็นระดับชั้นที่เทียบเท่ากับระดับชั้นย่อย MAC (MAC sublayer) ของมาตรฐาน LAN 802.3-802.5 ซึ่งเป็นระดับชั้นย่อยที่อยู่ส่วนล่างของระดับชั้นเชื่อมโยงข้อมูล (Datalink layer) ของ OSI (Open System Interconnection) โดยระดับชั้น DQDB นี้จะสามารถให้บริการแก่ระดับชั้นที่อยู่สูงกว่าได้หลายประการ ดังนี้ คือ

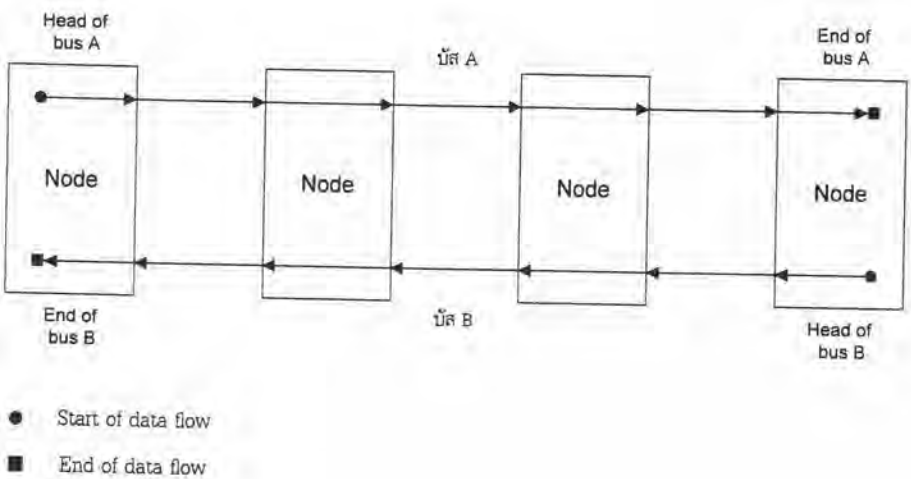
- ให้บริการ connectionless (datagram) MAC แก่ระดับชั้นย่อย IEEE 802.2 LLC ซึ่งเป็นบริการแบบเดียวกับของ IEEE 802 LAN และ FDDI (Fiber Distributed Data Interface)
- ให้บริการ connection-oriented (virtual circuit)
- ให้บริการ isochronous (circuit switched)

โดย MAC, Connection-oriented และ Isochronous Convergence Function จะทำหน้าที่แมปข้อมูลจากระดับชั้นที่สูงกว่าให้เป็นข้อมูลขนาด 53 ไบต์ที่ใช้ในระดับชั้น DQDB โดยระดับชั้น DQDB จะสนับสนุนการเข้าถึงตัวกลาง 2 วิธี คือ queued arbitrated (QA) โดยจะใช้สำหรับบริการที่เรียกว่า "non-time-sensitive" เช่น การส่งข้อมูลแบบอะซิงโครนัส หรือ การส่งข้อมูลแบบสวิตช์กลุ่มข้อมูล และแบบ pre arbitrated (PA) โดยจะใช้สำหรับบริการที่เรียกว่า "time-sensitive" เช่น การส่งข้อมูลแบบสวิตช์วงจร

2.3 องค์ประกอบของโครงข่าย DQDB

DQDB จะใช้โทโปโลยีแบบบัสคู่ ซึ่งประกอบด้วยบัสที่ส่งข้อมูลได้ทิศทางเดียว 2 เส้น และโนดต่าง ๆ ที่เชื่อมต่อเข้ากับบัสทั้งสองเส้น

2.3.1 สถาปัตยกรรมบัสคู่ (Dual bus architecture)



รูปที่ 2.3 สถาปัตยกรรมบัสคู่

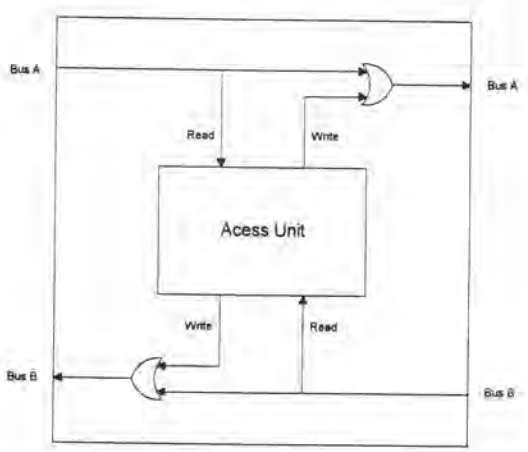
รูปที่ 2.3 แสดงถึงสถาปัตยกรรมบัสคู่ โดยบัสทั้งสองเส้น คือ Bus A และ Bus B จะส่งข้อมูลในทิศทางตรงข้าม ทำให้โนดคู่ใด ๆ ทำการติดต่อสื่อสารแบบ Full duplex ได้ อย่างไรก็ตาม โนดใดๆ ที่ต้องการส่งข้อมูลจำเป็นจะต้องรู้ว่าจะส่งข้อมูลไปยังโนดปลายทางโดยใช้บัสเส้นใด มิฉะนั้นโนดนั้นก็จำเป็นที่จะต้องส่งข้อมูลออกไปยังบัสทั้งสอง เพื่อที่จะมั่นใจได้ว่าสามารถส่งข้อมูลไปยังโนดปลายทางได้

บัส A และบัส B จะทำงานเป็นอิสระต่อกัน และเนื่องจากทุกโนดจะต้องต่อเข้ากับบัสทั้งสองเส้น และบัสทั้งสองเส้นก็มีการทำงานตลอดเวลา จึงทำให้ความจุช่องสัญญาณของโครงข่าย DQDB มีค่าเป็น 2 เท่าของอัตราข้อมูลของบัสแต่ละเส้น

การส่งข้อมูลของแต่ละบัสจะส่งผ่านหน่วยข้อมูลที่มีขนาดคงที่ที่เรียกว่า "สลอต (slot)" โดยโนดต่างๆ จะเขียนข้อมูลลงในสลอต ซึ่งทั้งนี้การที่โนดใด ๆ จะเขียนข้อมูล จะต้องขึ้นกับวิธีการเข้าถึงด้วย สำหรับทุก ๆ สลอตจะถูกสร้างโดย Head of bus และถูกนำออกจากบัสเมื่อมาถึงโนดที่เรียกว่า End of bus ซึ่งจากรูปที่ 2.3 แม้ว่าโครงข่าย DQDB จะมีรูปแบบทางกายภาพเป็นแบบวงแหวน แต่รูปแบบทางตรรกจะเป็นแบบบัส โดยที่บัส A และบัส B จะแยกออกจากกันและเป็นอิสระต่อกัน และสลอตของอีกบัสหนึ่งจะไม่ถูกส่งผ่านออกไปยังบัสอื่น

2.3.2 โนดของโปรโตคอล DQDB (DQDB nodes)

แต่ละโนดจะประกอบไปด้วย access unit (AU) ซึ่งจะถูกเชื่อมต่อกับบัสทั้งสองเส้นโดย AU จะทำหน้าที่สำคัญ 2 ประการ คือ ทำหน้าที่ในระดับชั้น DQDB และทำหน้าที่เชื่อมต่อเข้ากับบัสแต่ละเส้นเพื่ออ่านและเขียนข้อมูลลงในบัส ดังรูปที่ 2.4



รูปที่ 2.4 โนดของโปรโตคอล DQDB

โดย AU จะเขียนข้อมูลในบัสโดยใช้วิธีที่เรียกว่า "OR-write" ซึ่งจะเป็นการดำเนินการทางตรรกแบบ OR ระหว่างบิตที่อยู่บนบัสกับบิตที่โนดนั้นต้องการจะส่ง เช่น ถ้าต้องการที่จะเขียนบิต 0 ลงไปในบัส ผลลัพธ์ก็จะเป็นค่าทางตรรกที่อยู่บนบัส แต่ถ้าต้องการเขียน บิต 1 ลงไปในบัส ผลลัพธ์บนบัสก็จะเป็นบิต 1

ส่วนที่ทำหน้าที่อ่านข้อมูลจะอยู่ก่อนหน้าส่วนที่ทำหน้าที่เขียนข้อมูล เพราะจะทำให้ข้อมูลที่เข้ามานั้นไม่ถูกเขียนทับโดยข้อมูลที่จะส่ง ก่อนที่จะถูกอ่านเข้ามา

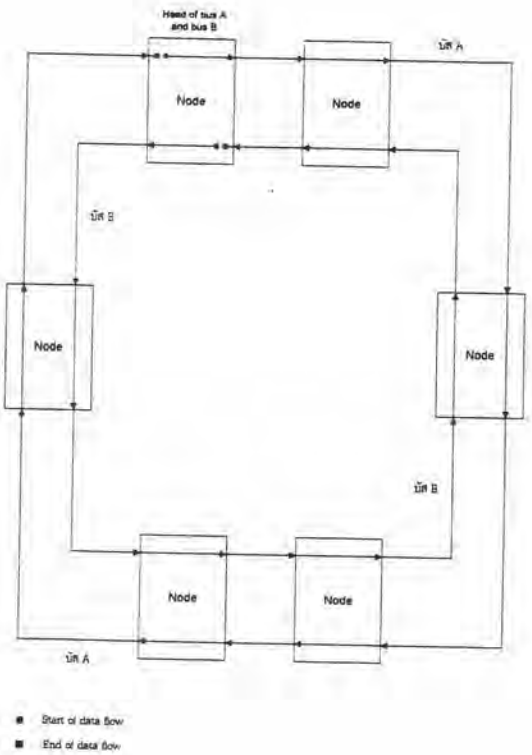
ส่วนที่ทำหน้าที่อ่านและเขียนข้อมูลนั้นอาจเป็นแบบ active หรือ passive ก็ได้ โดยในแบบ active นั้นจะอ่านข้อมูลเข้ามาทางบัสขาเข้า หลังจากนั้นจะทำการสร้างสัญญาณขึ้นมาใหม่แล้วส่งข้อมูลออกไปทางบัสขาออก ส่วนวิธีแบบ passive นั้นข้อมูลจะถูกอ่านโดยการดึงสัญญาณบางส่วนออกมาจากบัส ส่วนการเขียนก็จะส่งสัญญาณลงไปในบัสโดยตรง ซึ่งจะเห็นว่าวิธีแบบ passive นั้นจะจำกัดจำนวนโนดบนบัส เพราะว่าการที่สัญญาณต้องผ่านไปยังแต่ละโนดนั้นจะต้องถูกอ่าน ซึ่งจะทำให้เกิดการสูญเสียของสัญญาณ

คุณสมบัติที่สำคัญอีกประการหนึ่งของ DQDB ก็คือ การทำงานของบัสจะเป็นอิสระจากการทำงานของแต่ละ AU ดังนั้น เมื่อ AU หยุดทำงานจึงไม่ส่งผลต่อการทำงานโครงข่าย

โนดที่อยู่ส่วนหัวของบัสจะต้องทำหน้าที่พิเศษที่เรียกว่า "Head of Bus (HOB) Function" ซึ่งหน้าที่ที่สำคัญของ HOB คือ การสร้างสลอตว่าง เพื่อให้โนดต่าง ๆ ใช้ในการส่งข้อมูล

2.3.3 ความทนต่อความผิดพร่องของโปรโตคอล DQDB (DQDB Fault tolerance)

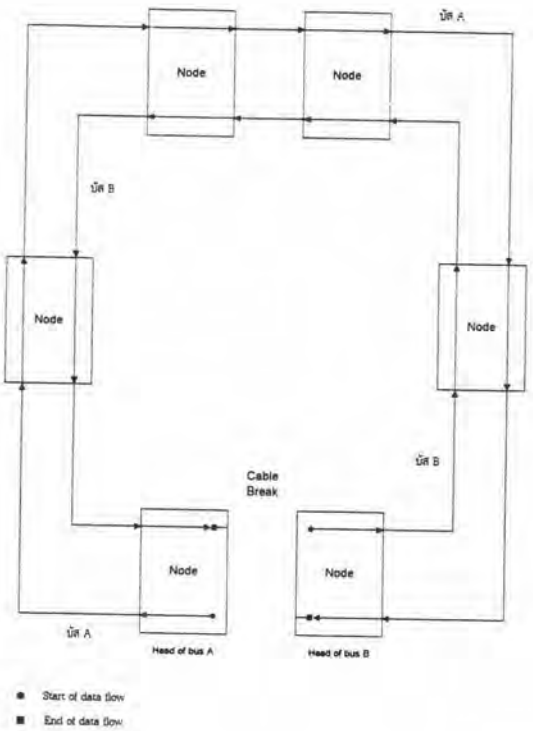
คุณลักษณะที่สำคัญประการหนึ่งของ DQDB ก็คือ ความสามารถที่เรียกว่า "self - healing" ซึ่งจะทำให้โครงข่ายสามารถที่จะทำงานได้เมื่อเกิดความผิดพลาดขึ้น ซึ่งในกรณีของโทโปโลยีแบบบัสคู่แบบวง (looped dual bus) ดังรูปที่ 2.5 ซึ่งโทโปโลยีแบบนี้คล้ายกับแบบโทโปโลยีบัสคู่แบบเปิด (open dual bus) ยกเว้นเพียงแต่จะมีเพียงโนดเดียวที่จะทำหน้าที่เป็นทั้ง HOB และ EOB ของบัสทั้งสองเส้น



รูปที่ 2.5 โทโปโลยีบัสคู่แบบวง

ในกรณีที่สายเคเบิลขาด หรือมีโนดที่ใช้งานไม่ได้ โครงข่ายจะทำการปรับตัวแบบอัตโนมัติให้เป็นแบบบัสคู่แบบเปิด โดยการย้ายหน้าที่ของ HOB ไปให้ยังโนดที่อยู่ใกล้กับจุดที่เกิดความผิดพลาด ดังรูปที่ 2.6 ซึ่งในกรณีนี้โนดต่าง ๆ ที่ยังคงปฏิบัติการได้จะยังคงเป็นส่วนหนึ่งของโครงข่าย

แต่ถ้าหากโนดที่อยู่ใกล้จุดที่เกิดความผิดพลาดที่สุดไม่สามารถทำหน้าที่ HOB ได้ ก็ให้โนดที่อยู่ใกล้จุดที่เกิดความผิดพลาดที่สุดที่สามารถทำหน้าที่ HOB ได้นั้นทำหน้าที่นี้แทน ดังนั้นในกรณีนี้ โนดที่ไม่สามารถทำหน้าที่ HOB ดังกล่าวจะถูกแยกออกจากโครงข่าย



รูปที่ 2.6 การทนต่อความผิดพร่องของบัสคู่แบบวง

2.4 การควบคุมการเข้าถึง (Access control)

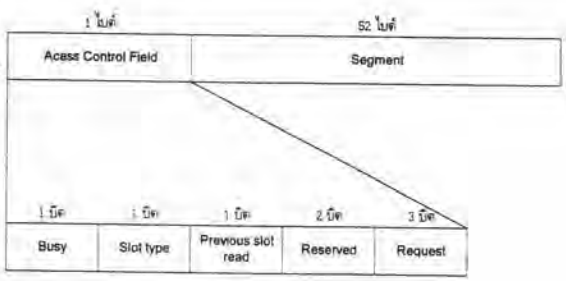
โครงข่าย DQDB นั้นสามารถให้บริการได้หลายรูปแบบ ได้แก่ การให้บริการข้อมูล เช่น connectionless MAC และ connection-oriented เป็นต้น ซึ่งบริการเหล่านี้จะเข้าถึงตัวกลางก็ต่อเมื่อมีข้อมูลที่จะส่งเท่านั้น ดังนั้นบริการเหล่านี้จึงไม่จำเป็นต้องจองเวลาในการใช้ตัวกลางแต่จะใช้การจองตัวกลางตามความต้องการ ซึ่งบริการเหล่านี้จะใช้ช่องสัญญาแบบ QA (Queued Arbitrated)

ส่วนบริการแบบ Isochronous เช่น เสียง และสัญญาณวิดีโอ จำเป็นจะต้องเข้าถึงตัวกลางอย่างสม่ำเสมอ ดังนั้นจึงใช้การเข้าถึงของสัญญาณแบบ PA (Pre Arbitrated)

2.4.1 วิธีการเข้าถึงแบบ QA (Queued arbitrated access method)

วิธีการเข้าถึงตัวกลางแบบ QA สำหรับบริการข้อมูลนั้น จะใช้คิวแบบกระจาย (distributed queue) ซึ่งวิธีการนี้จะค่อนข้างตรงไปตรงมาแต่ก็ซับซ้อนกว่าวิธีการของ CSMA/CD หรือ token passing โดยในส่วนนี้จะกล่าวถึงวิธีการทำงานของคิวแบบกระจาย

วิธีการเข้าถึงตัวกลางแบบ QA จะเหมาะกับบริการที่เป็นแบบเบิรสต์ ซึ่งจะมีข้อมูลที่ส่งเป็นจำนวนมากภายในเวลาที่จำกัด ซึ่งการทำงานของโปรโตคอลนี้จะอาศัย 2 ฟิลด์ของ ACF ซึ่งจะอยู่ในส่วนหัว (header) ของสลอตแบบ QA ดังรูปที่ 2.7



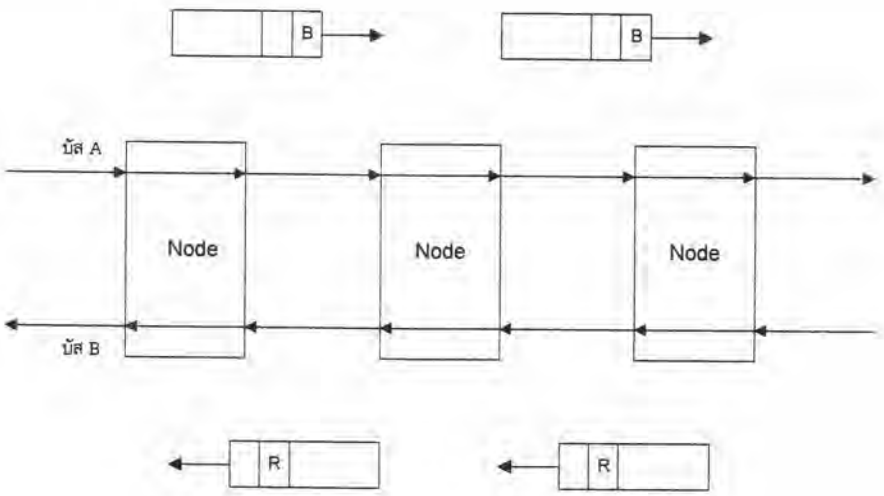
รูปที่ 2.7 รูปแบบของสลอต และส่วน ACF

โดยบิต Busy จะเป็นตัวบอกว่าสลอตนั้นว่างหรือไม่ ส่วนบิต Request จะใช้สำหรับการจองการเข้าถึงตัวกลาง โดยในส่วนต่อจากนี้จะเป็นการอธิบายถึงการทำงานของ QA ซึ่งเราจะสมมติให้มีเพียง 2 โนดที่อยู่บนบัส และโนด i จะอยู่ upstream จากโนด j กล่าวคือ สลอตจะมาถึงโนด i ก่อนที่จะมาถึงโนด j

เมื่อโนด i มีข้อมูลที่จะส่งออกไปยังโนด j โนด i จำเป็นจะต้องรู้ว่าจะส่งออกไปยังบัสใด ซึ่งในตัวอย่างนี้โนด i จะส่งข้อมูลไปยังโนด j โดยใช้บัส A ซึ่งโนด i จะต้องทำการจองสลอต QA

โดยการเซตค่าบิต Request ที่ว่างอยู่บนบัส B ซึ่งในกรณีนี้ทุกโนดที่อยู่ downstream จากโนด i บนบัส B จะได้รับ Request นี้ และเนื่องจากโนดที่อยู่ downstream ของโนด i บนบัส B จะอยู่ upstream จากโนด i บนบัส A ดังนั้นโนด i จึงได้เข้าสู่คิวบนบัส A ตามต้องการ

ทุกโนดที่อยู่ upstream จากโนด i บนบัส A จะทราบถึงจำนวนของการ Request ที่อยู่ก่อนหน้าในคิวได้จาก counter ที่โนดนั้น โดย counter ในแต่ละโนดจะบอกถึงจำนวนของการ Request ที่อยู่ก่อนหน้าในคิว ซึ่งก็คือจำนวนสลอตว่างที่โนดนั้นต้องปล่อยผ่านไปก่อนที่ตัวมันจะเข้าถึงตัวกลางได้ เมื่อมีสลอตว่างผ่านมา โนดนั้นจะลดค่า counter ลง 1 ซึ่งเมื่อ counter มีค่าเป็น 0 โนดนั้นจะสามารถเขียนข้อมูลลงในสลอตว่างอันถัดไป

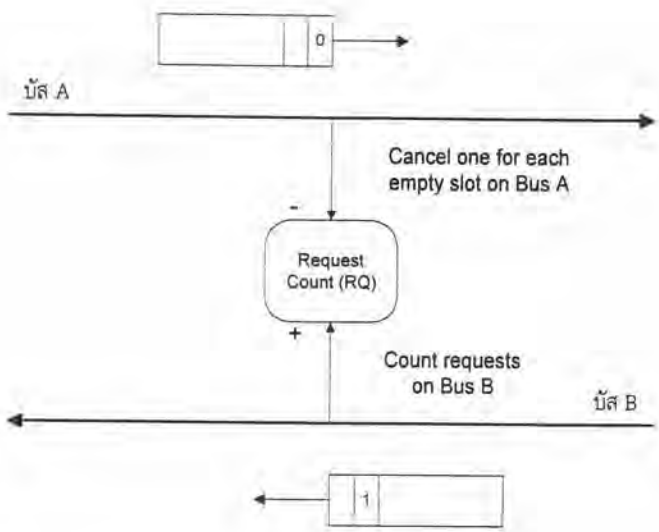


B = บิต Busy
R = บิต Request

รูปที่ 2.8 การทำงานของวิธีคิวแบบกระจายสำหรับการเข้าถึงบัส A

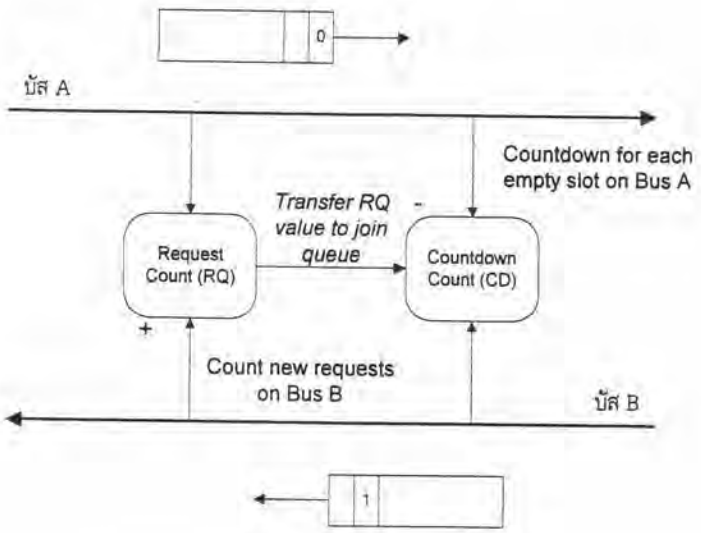
รูปที่ 2.8 แสดงการทำงานของวิธีคิวแบบกระจายสำหรับการเข้าถึงบัส A คำว่า forward bus ก็คือบัสที่เราต้องการจะส่งข้อมูลส่วน reverse bus คือบัสที่เราใช้ในการจองการเข้าถึงตัวกลาง ซึ่งในตัวอย่างนี้ forward bus ก็คือบัส A ส่วน reverse bus ก็คือ บัส B

แต่ละโนดจะนับจำนวนของการ Request ที่ได้เข้าคิวได้แล้วบน forward bus โดยการนับจำนวนของบิต Request ที่ถูกเซตค่าบน reverse bus ซึ่ง counter นี้จะบอกถึงความยาวของคิว เมื่อมีสลอตว่างผ่านมายัง forward bus จะหมายถึงว่าโนดที่อยู่ในคิวกำลังจะได้รับบริการ ดังนั้นค่า counter ซึ่งแสดงความยาวของคิวจะต้องลดลงไป 1 โดย request counter (RQ) จะใช้สำหรับนับความยาวของคิว และ RQ จะเพิ่มค่าอีก 1 เมื่อบิต Request ที่ผ่านมายัง reverse bus ถูกเซตค่าและ RQ จะมีค่าลดลงไป 1 เมื่อมีสลอตว่าผ่านมายัง forward bus ดังรูปที่ 2.9



รูปที่ 2.9 การทำงานของ request counter

แต่เมื่อโนดนั้นต้องการที่จะส่งข้อมูล มันจำเป็นจะต้องใช้ counter อีก 1 ตัวที่เรียกว่า countdown counter (CD) เมื่อโนดนั้นพร้อมที่จะส่งการ Request เพื่อที่จะเข้าคิว มันจะเซตค่าของ CD ให้เท่ากับค่าของ RQ และหลังจากนั้นจะเซตค่าของ RQ ให้เท่ากับ 0 ซึ่ง RQ ก็จะทำการนับจำนวนของ Request ต่อไป ส่วน CD จะทำหน้าที่บอกตำแหน่งของโนดนั้นในคิว ดังรูปที่ 2.10 โดยเมื่อค่าของ CD เท่ากับ 0 โนดนั้นจะสามารถส่งข้อมูลได้เมื่อมีสลอตว่างอันถัดไปมาถึง ในขณะที่รอคอยให้ค่า CD เป็น 0 เพื่อจะส่งข้อมูล RQ ก็จะทำการนับค่าของ Request อันใหม่ไปด้วย ซึ่งจะทำให้โนดนั้นทราบถึงจำนวนของโนดที่อยู่ในคิวก่อนหน้าพร้อมทั้งความยาวของคิวที่อยู่ข้างหลังมันด้วย



รูปที่ 2.10 การทำงานของ countdown counter

2.4.2 วิธีการเข้าถึงแบบ PA (Pre-arbitrated access method)

วิธีการเข้าถึงแบบ PA จะใช้สำหรับส่งข้อมูลที่เป็นแบบ isochronous ซึ่งการเข้าถึงแบบ PA จะแตกต่างกับแบบ QA มาก ซึ่งสลอตแบบ PA ก็จะมีขนาดคงที่เช่นเดียวกับสลอตแบบ QA แต่ทุกไบต์ของสลอตแบบ QA จะเป็นของโนดใดโนดหนึ่งโดยเฉพาะในขณะที่แต่ละไบต์ในสลอตของ PA จะแบ่งกันใช้ระหว่างโนดต่าง ๆ

เมื่อ HOB สร้างสลอตแบบ PA มันก็จะใส่ VCI (Virtual Channel Identifier) ไว้ที่ส่วนหัวของสลอต แต่ละโนดก็จะมีตารางที่บอกว่า ในแต่ละค่าของ VCI โนดนั้นจะรับและส่งข้อมูลได้ที่ไบต์ใด ทุกโนดจะต้องตรวจสอบค่า VCI ที่อยู่ในส่วนหัวของสลอตแบบ PA ถ้าค่า VCI ตรงกับที่มีอยู่ในตาราง มันก็จะอ่านหรือเขียนข้อมูลลงในสลอตตามตำแหน่งของไบต์ที่ระบุในตาราง แต่ถ้าตรวจสอบแล้วค่าของ VCI ไม่มีในตารางโนดนั้นก็จะไม่สนใจสลอต PA นี้ และปล่อยผ่านไปยังโนดอื่น ๆ ที่อยู่ downstream

รายละเอียดของโปรโตคอล DQDB สามารถศึกษาเพิ่มเติมได้จาก [1-3]