

บทที่ 5

ผลการจำลองแบบ และวิเคราะห์ผลการจำลองแบบ

5.1 ค่าย่อ

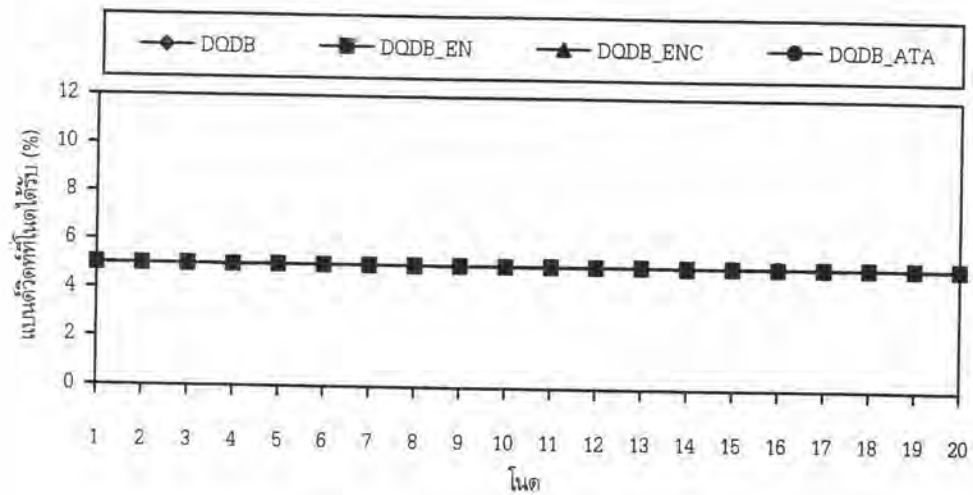
| | | |
|-------------------|---|--|
| Bw_i | = | ค่าแบนด์วิดท์ที่โนด i ได้รับ โดยจะแสดงเป็นเปอร์เซ็นต์ของแบนด์วิดท์ทั้งหมดของโครงข่าย |
| \overline{Bw}_1 | = | ค่าแบนด์วิดท์เฉลี่ยของโนดที่ 1-10 |
| \overline{Bw}_2 | = | ค่าแบนด์วิดท์เฉลี่ยของโนดที่ 11-20 |
| \overline{Bw}_3 | = | ค่าแบนด์วิดท์เฉลี่ยของโนดที่ 1-20 |
| Dev_1 | = | ส่วนเบี่ยงเบนของแบนด์วิดท์ของโนดที่ 1-10 |
| Dev_2 | = | ส่วนเบี่ยงเบนของแบนด์วิดท์ของโนดที่ 11-20 |
| Dev_3 | = | ส่วนเบี่ยงเบนของแบนด์วิดท์ของโนดที่ 1-20 |
| DQDB | = | โปรโตคอล DQDB ที่ไม่มีการใช้โนดอีเรเซอร์ |
| DQDB_EN | = | โปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์แบบมูลฐาน |
| DQDB_ENC | = | โปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์ที่ใช้ตัวนับ |
| DQDB_ATA | = | โปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรชโฮลด์ที่โนดอีเรเซอร์แบบปรับตัว |

5.2 ผลการจำลองแบบ

ในส่วนนี้จะเป็นผลที่ได้จากการจำลองแบบของโครงข่ายในรูปที่ 4.1 ที่ใช้โปรโตคอล DQDB ที่ไม่มีการใช้โนดอีเรเซอร์, โปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์แบบมูลฐาน, โปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์ที่ใช้ตัวนับ, และโปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรชโฮลด์ที่โนดอีเรเซอร์แบบปรับตัว ซึ่งจะแสดงเป็นเปอร์เซ็นต์ของแบนด์วิดท์ที่โนดต่าง ๆ ได้รับ พร้อมทั้งเส้นแสดงแนวโน้มของแบนด์วิดท์ เมื่อได้ทำการเปลี่ยนแปลงค่าโหลดจาก 100% ไปเป็น 120%, 140%, 160%, 180%, และ 200% ตามลำดับ

ตารางที่ 5.1 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 100%

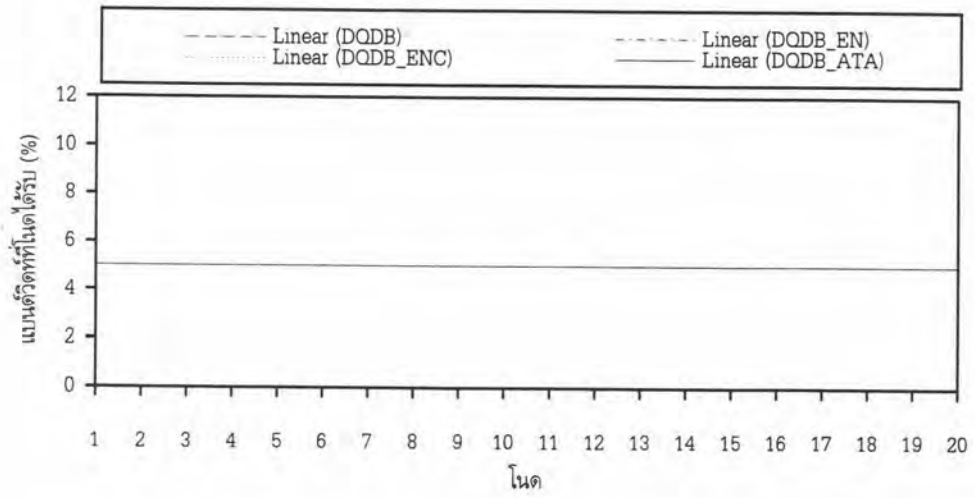
| Node index | Bandwidth Allocation (%) | | | |
|---|--------------------------|---------|----------|----------|
| | DQDB | DQDB_EN | DQDB_ENC | DQDB_ATA |
| 1 | 4.99 | 5.00 | 5.00 | 5.00 |
| 2 | 5.00 | 4.99 | 5.00 | 5.00 |
| 3 | 5.00 | 5.00 | 5.01 | 4.99 |
| 4 | 4.97 | 4.99 | 5.00 | 4.99 |
| 5 | 5.01 | 5.00 | 5.00 | 5.00 |
| 6 | 5.01 | 4.99 | 5.01 | 5.00 |
| 7 | 5.01 | 5.00 | 4.99 | 5.00 |
| 8 | 5.01 | 5.00 | 4.99 | 4.99 |
| 9 | 4.97 | 5.00 | 5.00 | 5.02 |
| 10 | 5.02 | 4.99 | 5.00 | 4.98 |
| 11 | 5.01 | 5.02 | 5.00 | 4.98 |
| 12 | 5.02 | 5.00 | 5.01 | 4.99 |
| 13 | 4.99 | 5.02 | 5.00 | 5.01 |
| 14 | 5.01 | 5.01 | 4.98 | 5.00 |
| 15 | 5.02 | 5.00 | 5.00 | 5.02 |
| 16 | 5.02 | 4.98 | 5.01 | 5.02 |
| 17 | 4.97 | 5.01 | 4.99 | 5.00 |
| 18 | 5.00 | 5.01 | 5.00 | 5.02 |
| 19 | 4.98 | 5.00 | 5.00 | 5.01 |
| 20 | 4.98 | 5.00 | 5.00 | 4.98 |
| $\overline{Bw}_1 = \frac{1}{10} \sum_{i=1}^{10} Bw_i$ | 5.00 | 5.00 | 5.00 | 5.00 |
| $\overline{Bw}_2 = \frac{1}{10} \sum_{i=11}^{20} Bw_i$ | 5.00 | 5.01 | 5.00 | 5.00 |
| $\overline{Bw}_3 = \frac{1}{20} \sum_{i=1}^{20} Bw_i$ | 5.00 | 5.00 | 5.00 | 5.00 |
| $Dev_1 = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (Bw_i - 5)^2}$ | 0.02 | 0.01 | 0.01 | 0.01 |
| $Dev_2 = \sqrt{\frac{1}{10} \sum_{i=11}^{20} (Bw_i - 5)^2}$ | 0.02 | 0.01 | 0.01 | 0.02 |
| $Dev_3 = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (Bw_i - 5)^2}$ | 0.02 | 0.01 | 0.01 | 0.01 |



รูปที่ 5.1 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 100%

เมื่อนำผลที่ได้มาเขียนกราฟความสัมพันธ์ของแบนด์วิดท์ที่โหนดได้รับของโหนดต่าง ๆ ทั้ง 20 โหนด ที่โหลดเท่ากับ 100% จะได้กราฟดังรูปที่ 5.1 โดยจะเห็นว่า ในทุกวิธี โหนดต่าง ๆ จะได้รับแบนด์วิดท์ในปริมาณที่เท่า ๆ กัน ซึ่งแบนด์วิดท์ที่แต่ละโหนดได้รับจะเท่ากับ $(100\%) / (20 \text{ โหนด})$ หรือคิดเป็น 5% ของแบนด์วิดท์ทั้งหมด

ในรูปที่ 5.2 จะแสดงเส้นแนวโน้มของแบนด์วิดท์ที่โหนดต่าง ๆ จะได้รับ เมื่อโหลดเท่ากับ 100% ซึ่งจะเห็นว่า เส้นทั้งหมดจะเป็นเส้นตรงที่ขนานกับแกนในแนวราบ แสดงว่า ในทุกวิธี โหนดต่าง ๆ จะมีแนวโน้มที่จะได้รับแบนด์วิดท์เท่ากัน

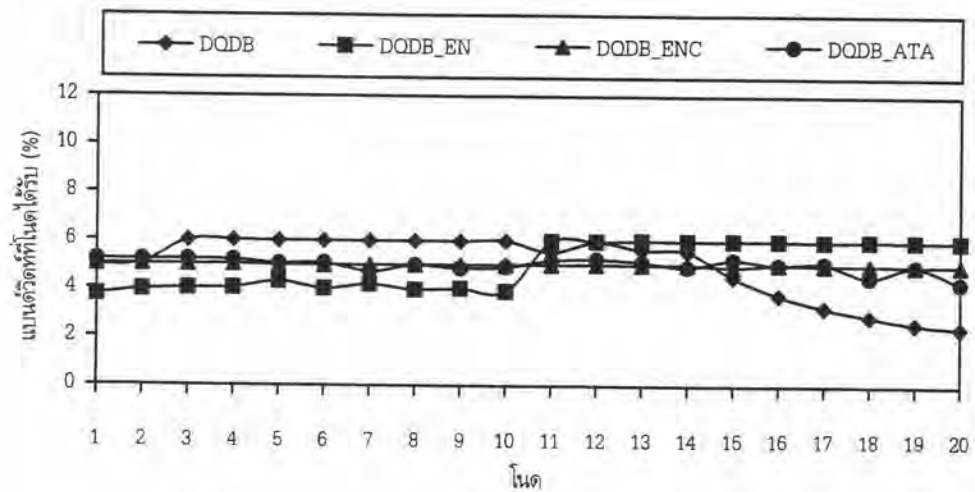


รูปที่ 5.2 เส้นแนวโน้มของแบนด์วิดท์ที่โน้ดได้รับ เมื่อโหลดเท่ากับ 100%

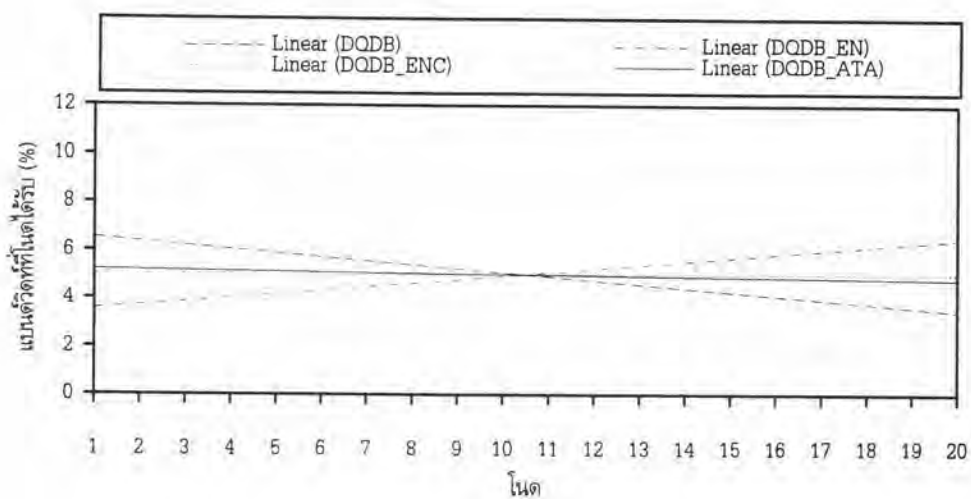
เมื่อทำการจำลองแบบโดยเปลี่ยนค่าของโหลดไปเป็น 120% จะได้ผลดังตารางที่ 5.2

ตารางที่ 5.2 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 120%

| Node index | Bandwidth Allocation (%) | | | |
|---|--------------------------|---------|----------|----------|
| | DODB | DODB_EN | DODB_ENC | DODB_ATA |
| 1 | 4.94 | 3.72 | 5.01 | 5.20 |
| 2 | 5.04 | 3.94 | 5.01 | 5.20 |
| 3 | 6.00 | 4.01 | 5.00 | 5.21 |
| 4 | 6.01 | 4.02 | 5.02 | 5.20 |
| 5 | 5.99 | 4.29 | 4.99 | 5.05 |
| 6 | 5.99 | 4.00 | 4.98 | 5.07 |
| 7 | 5.99 | 4.17 | 4.99 | 4.72 |
| 8 | 5.99 | 3.96 | 5.01 | 4.99 |
| 9 | 5.99 | 4.01 | 5.01 | 4.86 |
| 10 | 6.00 | 3.88 | 5.01 | 4.91 |
| 11 | 5.54 | 5.98 | 5.00 | 5.20 |
| 12 | 5.97 | 5.98 | 5.02 | 5.25 |
| 13 | 5.63 | 6.02 | 5.00 | 5.13 |
| 14 | 5.56 | 6.00 | 5.02 | 4.92 |
| 15 | 4.51 | 6.01 | 4.93 | 5.24 |
| 16 | 3.75 | 6.02 | 5.03 | 5.02 |
| 17 | 3.24 | 5.99 | 5.00 | 5.09 |
| 18 | 2.88 | 6.01 | 5.01 | 4.50 |
| 19 | 2.58 | 6.00 | 4.99 | 4.96 |
| 20 | 2.41 | 5.99 | 4.99 | 4.27 |
| $\overline{Bw}_1 = \frac{1}{10} \sum_{i=1}^{10} Bw_i$ | 5.79 | 4.00 | 5.00 | 5.04 |
| $\overline{Bw}_2 = \frac{1}{10} \sum_{i=11}^{20} Bw_i$ | 4.21 | 6.00 | 5.00 | 4.96 |
| $\overline{Bw}_3 = \frac{1}{20} \sum_{i=1}^{20} Bw_i$ | 5.00 | 5.00 | 5.00 | 5.00 |
| $Dev_1 = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (Bw_i - 5)^2}$ | 0.89 | 1.01 | 0.01 | 0.17 |
| $Dev_2 = \sqrt{\frac{1}{10} \sum_{i=11}^{20} (Bw_i - 5)^2}$ | 1.55 | 1.00 | 0.03 | 0.31 |
| $Dev_3 = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (Bw_i - 5)^2}$ | 1.26 | 1.01 | 0.02 | 0.25 |



รูปที่ 5.3 แบนด์วิดท์ที่โน้ตได้รับ เมื่อโหลดเท่ากับ 120%



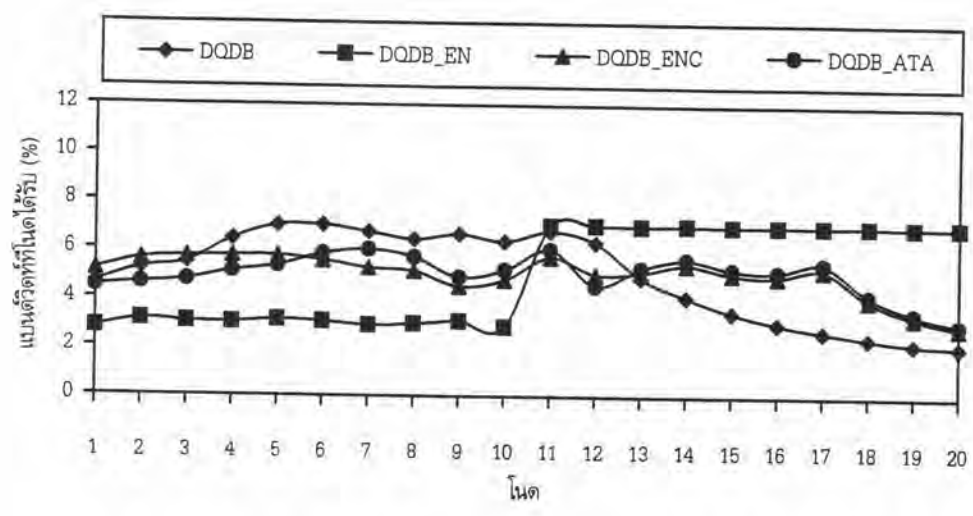
รูปที่ 5.4 เส้นแนวโน้มของแบนด์วิดท์ที่โน้ตได้รับ เมื่อโหลดเท่ากับ 120%

เมื่อพิจารณากราฟในรูปที่ 5.3 และ 5.4 ซึ่งแสดงค่าความสัมพันธ์ระหว่างแบนด์วิดท์ที่โน้ตได้รับกับหมายเลขโน้ต และเส้นแนวโน้มของแบนด์วิดท์ที่โน้ตจะได้รับ ที่โหลดเท่ากับ 120% ตามลำดับ จะเห็นว่า แบนด์วิดท์ที่โน้ตต่าง ๆ จะได้รับในแต่ละวิธีจะแตกต่างกัน โดยวิธีที่ใช้

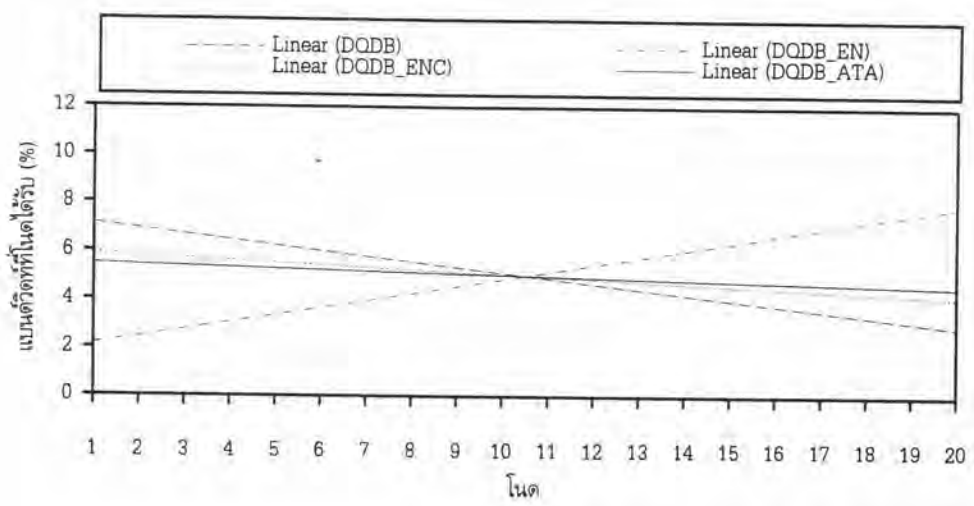
โปรโตคอล DQDB ที่ไม่มีการใช้โนดอีเรเซอร์นั้น โหนดที่อยู่ส่วนปลายของบัสจะได้รับแบนด์วิดท์น้อยกว่าโหนดที่อยู่ส่วนต้นของบัสค่อนข้างมากเพราะว่า โหนดที่อยู่ส่วนต้นของบัสสามารถใช้สล็อตว่างได้ก่อนทำให้เหลือสล็อตว่างที่จะผ่านไปยังส่วนปลายของบัสน้อยลง จึงทำให้โหนดที่อยู่ส่วนปลายของบัสส่งข้อมูลได้น้อยลง ส่วนในโปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์แบบมูลฐานซึ่งไม่มีการยกเลิกการจองสล็อต โหนดที่อยู่ส่วนปลายของบัสจะได้รับแบนด์วิดท์มากกว่าโหนดที่อยู่ส่วนต้นของบัส ซึ่งเป็นเพราะว่า โหนดที่อยู่ส่วนปลายของบัสจะได้รับแบนด์วิดท์ที่เพิ่มขึ้นจากสล็อตที่ถูกลบโดยโนดอีเรเซอร์ ทำให้สามารถส่งข้อมูลได้มากขึ้น และในขณะเดียวกันก็จะทำให้เกิดการจองสล็อตมากขึ้นตามไปด้วย และเมื่อโหนดที่อยู่ส่วนต้นของบัสได้รับสัญญาณการจองสล็อตมากขึ้น มันก็จะปล่อยสล็อตว่างให้ผ่านไปมากขึ้น จึงทำให้โหนดเหล่านี้ส่งข้อมูลได้น้อยลง และด้วยสาเหตุนี้เองจึงจำเป็นต้องมีกระบวนการในการยกเลิกการจองสล็อต สำหรับในโปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์ที่ใช้ตัวนับ และโปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรซโฮลด์ที่โนดอีเรเซอร์แบบปรับตัวนั้น จะให้ผลที่ดีมากกว่า 2 วิธีข้างต้น โดยโปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรซโฮลด์ที่โนดอีเรเซอร์แบบปรับตัวนั้น โหนดที่อยู่ปลาย ๆ ของบัสจะได้รับแบนด์วิดท์น้อยลงเล็กน้อย

ตารางที่ 5.3 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 140%

| Node index | Bandwidth Allocation (%) | | | |
|---|--------------------------|---------|----------|----------|
| | DQDB | DQDB_EN | DQDB_ENC | DQDB_ATA |
| 1 | 4.63 | 2.79 | 5.17 | 4.49 |
| 2 | 5.24 | 3.12 | 5.62 | 4.63 |
| 3 | 5.49 | 3.04 | 5.73 | 4.76 |
| 4 | 6.43 | 3.02 | 5.76 | 5.13 |
| 5 | 6.99 | 3.12 | 5.75 | 5.34 |
| 6 | 7.01 | 3.05 | 5.57 | 5.83 |
| 7 | 6.74 | 2.90 | 5.27 | 6.02 |
| 8 | 6.44 | 2.97 | 5.14 | 5.72 |
| 9 | 6.65 | 3.08 | 4.51 | 4.90 |
| 10 | 6.35 | 2.85 | 4.78 | 5.20 |
| 11 | 6.78 | 7.01 | 5.73 | 6.03 |
| 12 | 6.33 | 7.02 | 5.03 | 4.56 |
| 13 | 4.90 | 7.00 | 5.10 | 5.28 |
| 14 | 4.10 | 7.02 | 5.43 | 5.63 |
| 15 | 3.47 | 7.00 | 5.04 | 5.23 |
| 16 | 3.02 | 7.00 | 4.93 | 5.14 |
| 17 | 2.68 | 7.00 | 5.22 | 5.47 |
| 18 | 2.42 | 7.01 | 4.01 | 4.19 |
| 19 | 2.20 | 7.00 | 3.30 | 3.45 |
| 20 | 2.08 | 7.01 | 2.89 | 3.00 |
| $\overline{Bw}_1 = \frac{1}{10} \sum_{i=1}^{10} Bw_i$ | 6.20 | 2.99 | 5.33 | 5.20 |
| $\overline{Bw}_2 = \frac{1}{10} \sum_{i=11}^{20} Bw_i$ | 3.80 | 7.01 | 4.67 | 4.80 |
| $\overline{Bw}_3 = \frac{1}{20} \sum_{i=1}^{20} Bw_i$ | 5.00 | 5.00 | 5.00 | 5.00 |
| $Dev_1 = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (Bw_i - 5)^2}$ | 1.42 | 2.01 | 0.53 | 0.54 |
| $Dev_2 = \sqrt{\frac{1}{10} \sum_{i=11}^{20} (Bw_i - 5)^2}$ | 2.01 | 2.01 | 0.95 | 0.95 |
| $Dev_3 = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (Bw_i - 5)^2}$ | 1.74 | 2.01 | 0.77 | 0.77 |



รูปที่ 5.5 แบนด์วิดท์ที่โนดได้รับ เมื่อโหลดเท่ากับ 140%



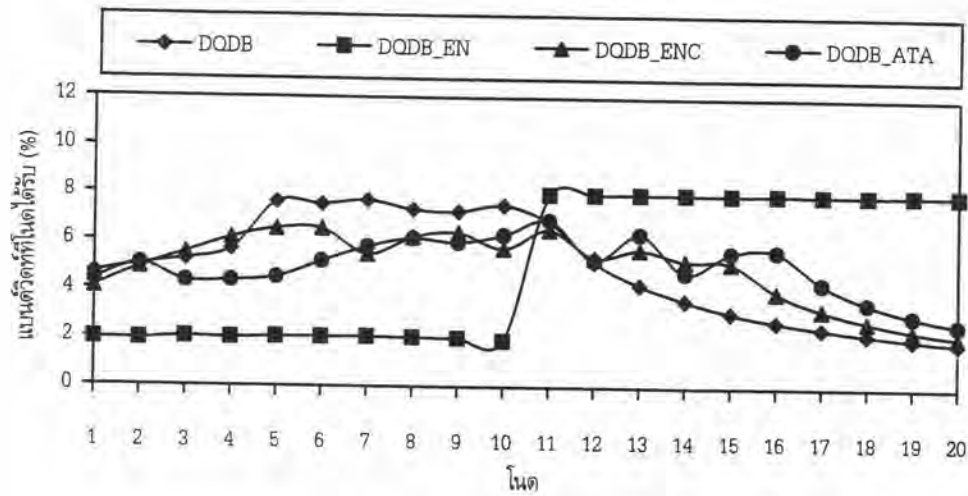
รูปที่ 5.6 เส้นแนวโน้มของแบนด์วิดท์ที่โนดได้รับ เมื่อโหลดเท่ากับ 140%

เมื่อได้เปลี่ยนค่าโหลดไปเป็น 140% กราฟจะเป็นดังรูปที่ 5.5 ซึ่งจะเห็นว่า เมื่อโหลดมีค่าสูงขึ้น แต่ละโนดจะได้รับแบนด์วิดท์แตกต่างกันมากขึ้น โดยเฉพาะในโปรโตคอล DQDB ที่ไม่มีการใช้โนดเอเรเซอร์ และโปรโตคอล DQDB ที่ใช้วิธีโนดเอเรเซอร์แบบมูลฐานนั้นจะเห็นความแตกต่าง

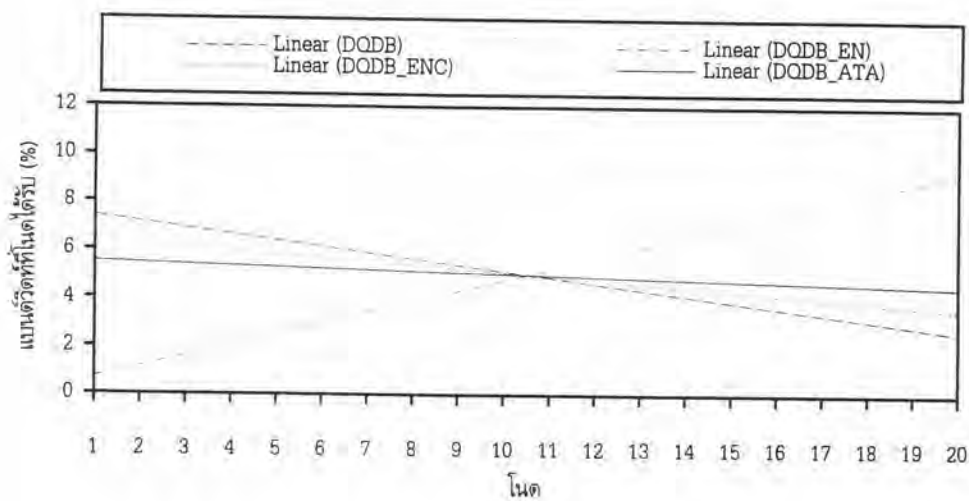
ต่างได้อย่างชัดเจน ส่วนในโปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์ที่ใช้ตัวนับ และโปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรชโฮลด์ที่โนดอีเรเซอร์แบบปรับตัวนั้นจะมีความแตกต่างกันน้อยกว่า และเมื่อเปรียบเทียบกันระหว่าง 2 วิธีนี้ ในโปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์ที่ใช้ตัวนับ โหนดที่อยู่ส่วนต้นของบัสจะได้รับแบนด์วิดท์มากกว่า แต่โหนดที่อยู่ส่วนปลายของบัสจะได้รับแบนด์วิดท์น้อยกว่าวิธีการปรับค่าเทรชโฮลด์ที่โนดอีเรเซอร์แบบปรับตัว และเมื่อพิจารณากราฟในรูปที่ 5.6 จะเห็นว่า วิธีการปรับค่าเทรชโฮลด์ที่โนดอีเรเซอร์แบบปรับตัวจะมีเส้นแนวโน้มของแบนด์วิดท์ที่ดีกว่าเล็กน้อย

ตารางที่ 5.4 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 160%

| Node index | Bandwidth Allocation (%) | | | |
|---|--------------------------|---------|----------|----------|
| | DQDB | DQDB_EN | DQDB_ENC | DQDB_ATA |
| 1 | 4.67 | 1.95 | 4.06 | 4.36 |
| 2 | 5.01 | 1.93 | 4.88 | 5.03 |
| 3 | 5.25 | 2.03 | 5.51 | 4.33 |
| 4 | 5.67 | 2.01 | 6.13 | 4.36 |
| 5 | 7.64 | 2.04 | 6.52 | 4.50 |
| 6 | 7.54 | 2.05 | 6.52 | 5.17 |
| 7 | 7.72 | 2.06 | 5.47 | 5.78 |
| 8 | 7.34 | 2.03 | 6.19 | 6.14 |
| 9 | 7.25 | 2.01 | 6.38 | 5.97 |
| 10 | 7.53 | 1.90 | 5.73 | 6.29 |
| 11 | 6.83 | 7.99 | 6.52 | 6.91 |
| 12 | 5.20 | 7.99 | 5.37 | 5.27 |
| 13 | 4.26 | 8.01 | 5.72 | 6.36 |
| 14 | 3.61 | 8.01 | 5.25 | 4.74 |
| 15 | 3.11 | 7.99 | 5.17 | 5.61 |
| 16 | 2.74 | 8.00 | 3.97 | 5.66 |
| 17 | 2.47 | 7.99 | 3.25 | 4.33 |
| 18 | 2.23 | 7.99 | 2.77 | 3.53 |
| 19 | 2.05 | 8.02 | 2.42 | 3.01 |
| 20 | 1.90 | 8.01 | 2.16 | 2.66 |
| $\overline{Bw}_1 = \frac{1}{10} \sum_{i=1}^{10} Bw_i$ | 6.56 | 2.00 | 5.74 | 5.19 |
| $\overline{Bw}_2 = \frac{1}{10} \sum_{i=11}^{20} Bw_i$ | 3.44 | 8.00 | 4.26 | 4.81 |
| $\overline{Bw}_3 = \frac{1}{20} \sum_{i=1}^{20} Bw_i$ | 5.00 | 5.00 | 5.00 | 5.00 |
| $Dev_1 = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (Bw_i - 5)^2}$ | 1.96 | 3.00 | 1.05 | 0.78 |
| $Dev_2 = \sqrt{\frac{1}{10} \sum_{i=11}^{20} (Bw_i - 5)^2}$ | 2.17 | 3.00 | 1.64 | 1.36 |
| $Dev_3 = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (Bw_i - 5)^2}$ | 2.07 | 3.00 | 1.38 | 1.11 |



รูปที่ 5.7 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 160%



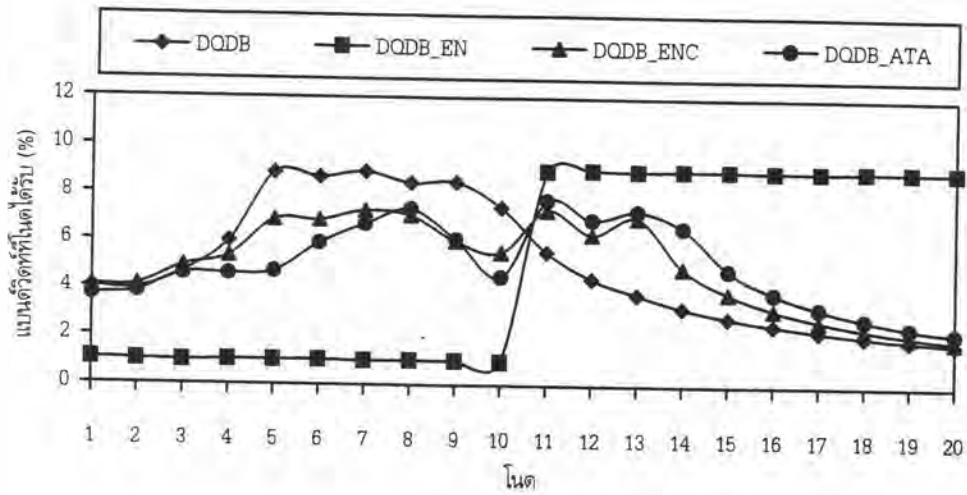
รูปที่ 5.8 เส้นแนวโน้มของแบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 160%

จากกราฟที่ 5.7 ซึ่งเป็นกรณีที่โหลดเป็น 160% ได้แสดงให้เห็นว่า โปรโตคอล DQDB ที่ใช้วิธีโหนดอีเรเซอร์ที่ใช้ตัวนับ และโปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรชโฮลด์ที่โหนดอีเรเซอร์แบบปรับตัวจะมีความแตกต่างกันมากขึ้น โดยจะเห็นได้ชัดเจนในบริเวณโหนดที่อยู่ส่วนต้นและส่วน

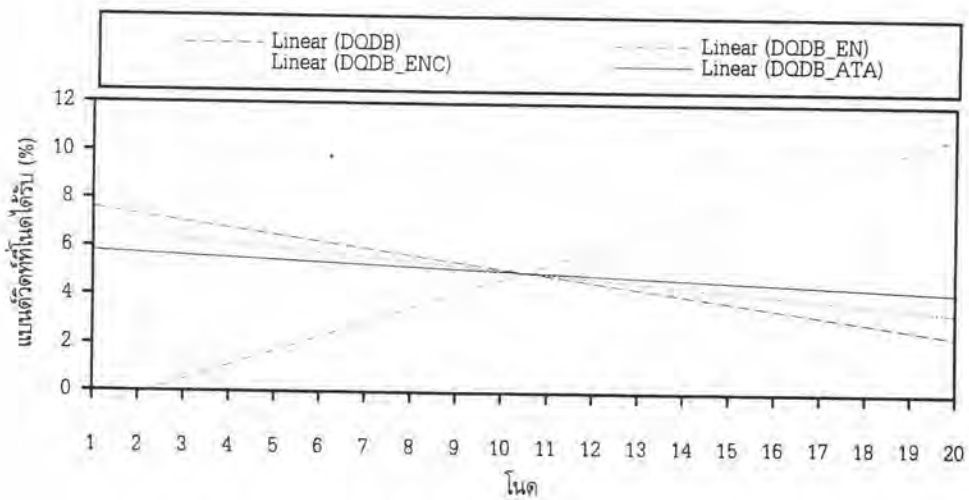
ปลายของบัส ซึ่งในกรณีนี้ โหนดที่อยู่ส่วนปลายของบัสของวิธีการปรับค่าเทรซโฮลด์ที่โนดอีเรเซอร์แบบปรับตัวจะได้รับแบนด์วิดท์มากกว่า แม้ว่าโหนดที่อยู่ส่วนต้นของบัสจะได้รับแบนด์วิดท์น้อยลง แต่โหนดเหล่านี้ก็ยังได้รับแบนด์วิดท์ที่ใกล้เคียง 5% (เป็นค่าแบนด์วิดท์ที่แต่ละโหนดจะได้รับ ในกรณีที่ ไม่เกิดสภาวะโหลตเกิน) ซึ่งหมายความว่า เราสามารถจัดสรรแบนด์วิดท์ส่วนเกินของทางต้นบัสไปยังปลายบัสที่ได้รับแบนด์วิดท์ค่อนข้างน้อยได้ดีขึ้น และในรูปที่ 5.8 เส้นแนวโน้มของวิธีการปรับค่าเทรซโฮลด์ที่โนดอีเรเซอร์แบบปรับตัวก็จะดีกว่าวิธีอื่น ๆ

ตารางที่ 5.5 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 180%

| Node index | Bandwidth Allocation (%) | | | |
|---|--------------------------|---------|----------|----------|
| | DQDB | DQDB_EN | DQDB_ENC | DQDB_ATA |
| 1 | 4.01 | 1.04 | 4.07 | 3.71 |
| 2 | 3.95 | 1.00 | 4.12 | 3.83 |
| 3 | 4.68 | 0.97 | 4.92 | 4.59 |
| 4 | 5.98 | 1.01 | 5.36 | 4.59 |
| 5 | 8.87 | 1.01 | 6.90 | 4.70 |
| 6 | 8.68 | 1.02 | 6.88 | 5.91 |
| 7 | 8.92 | 1.00 | 7.29 | 6.72 |
| 8 | 8.45 | 1.01 | 7.09 | 7.36 |
| 9 | 8.50 | 0.99 | 6.02 | 6.08 |
| 10 | 7.44 | 0.95 | 5.53 | 4.50 |
| 11 | 5.56 | 8.97 | 7.33 | 7.75 |
| 12 | 4.45 | 9.01 | 6.32 | 6.94 |
| 13 | 3.79 | 8.99 | 7.04 | 7.31 |
| 14 | 3.24 | 9.01 | 4.91 | 6.61 |
| 15 | 2.83 | 9.01 | 3.86 | 4.83 |
| 16 | 2.54 | 8.98 | 3.19 | 3.86 |
| 17 | 2.29 | 8.98 | 2.73 | 3.24 |
| 18 | 2.09 | 9.02 | 2.39 | 2.80 |
| 19 | 1.93 | 9.01 | 2.13 | 2.46 |
| 20 | 1.80 | 9.02 | 1.92 | 2.22 |
| $\overline{Bw}_1 = \frac{1}{10} \sum_{i=1}^{10} Bw_i$ | 6.95 | 1.00 | 5.82 | 5.20 |
| $\overline{Bw}_2 = \frac{1}{10} \sum_{i=11}^{20} Bw_i$ | 3.05 | 9.00 | 4.18 | 4.80 |
| $\overline{Bw}_3 = \frac{1}{20} \sum_{i=1}^{20} Bw_i$ | 5.00 | 5.00 | 5.00 | 5.00 |
| $Dev_1 = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (Bw_i - 5)^2}$ | 2.78 | 4.00 | 1.41 | 1.19 |
| $Dev_2 = \sqrt{\frac{1}{10} \sum_{i=11}^{20} (Bw_i - 5)^2}$ | 2.27 | 4.00 | 2.14 | 2.07 |
| $Dev_3 = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (Bw_i - 5)^2}$ | 2.54 | 4.00 | 1.81 | 1.69 |



รูปที่ 5.9 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 180%



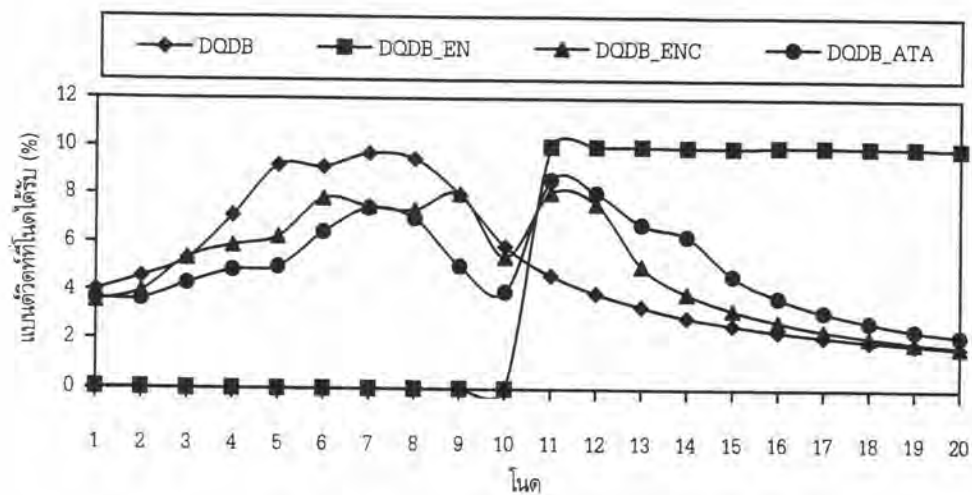
รูปที่ 5.10 เส้นแนวโน้มของแบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 180%

ที่โหลด 180% ผลที่ได้จะเป็นดังรูปที่ 5.9 ซึ่งจะเห็นว่า โปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทอร์ซโฮลด์ที่โหนดอีเรเซอร์แบบปรับตัวจะทำให้โหนดบริเวณปลายบัสได้รับแบนด์วิดท์มากกว่าของวิธีโหนดอีเรเซอร์ที่ใช้ตัวนับ และทำให้ค่าแบนด์วิดท์เฉลี่ยของโหนดเหล่านี้ (\overline{Bw}_2) เพิ่มจาก

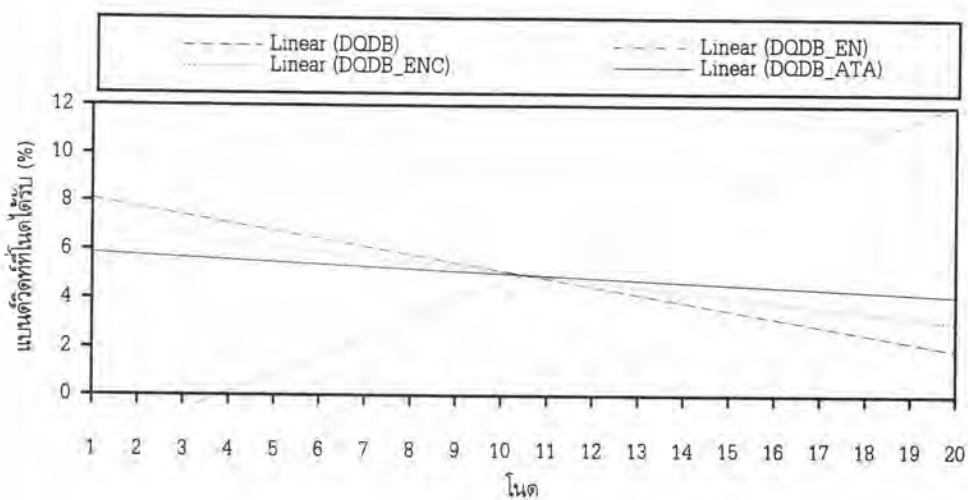
4.18% ไปเป็น 4.80% ซึ่งใกล้เคียงกับค่า 5% มากขึ้น โดยทั้งสองวิธีนั้น แบนด์วิดท์ที่โนดได้รับ จะมีค่าเพิ่มขึ้นในบริเวณช่วงต้นของบัส และจะมีค่าลดลงเมื่อถึงโนดที่อยู่ก่อนหน้าโนดอีเรเซอร์ (โนดที่ 10) และจะมีค่าเพิ่มขึ้นอีกครั้งที่โนดอีเรเซอร์ (โนดที่ 11) ซึ่งจากรูปจะเห็นว่า โหนดอีเรเซอร์ จะได้รับแบนด์วิดท์มากที่สุด เพราะว่า โหนดอีเรเซอร์นี้จะสามารถใช้สล็อตที่ถูกสลับได้ก่อนโนดอื่น ๆ และแบนด์วิดท์ที่โนดได้รับจะค่อย ๆ มีค่าลดลงเมื่อเข้าใกล้ปลายบัส โดยโนดที่ 20 ซึ่งเป็นโนดสุดท้ายจะได้รับแบนด์วิดท์น้อยที่สุด และในรูปที่ 5.10 จะเห็นว่า เส้นแนวโน้มของโปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรซโฮลด์ที่โนดอีเรเซอร์แบบปรับตัวจะดีกว่าวิธีอื่น ๆ โดยโนดที่อยู่บริเวณส่วนปลายของบัสจะมีแนวโน้มที่จะได้รับแบนด์วิดท์มากขึ้น

ตารางที่ 5.6 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 200%

| Node index | Bandwidth Allocation (%) | | | |
|---|--------------------------|---------|----------|----------|
| | DQDB | DQDB_EN | DQDB_ENC | DQDB_ATA |
| 1 | 4.01 | 0.04 | 3.55 | 3.67 |
| 2 | 4.58 | 0.03 | 3.98 | 3.66 |
| 3 | 5.32 | 0.02 | 5.36 | 4.31 |
| 4 | 7.12 | 0.02 | 5.88 | 4.85 |
| 5 | 9.21 | 0.02 | 6.24 | 4.99 |
| 6 | 9.15 | 0.02 | 7.84 | 6.44 |
| 7 | 9.71 | 0.02 | 7.52 | 7.43 |
| 8 | 9.48 | 0.02 | 7.36 | 7.01 |
| 9 | 8.01 | 0.02 | 8.05 | 5.03 |
| 10 | 5.85 | 0.02 | 5.41 | 3.97 |
| 11 | 4.67 | 9.99 | 8.08 | 8.62 |
| 12 | 3.92 | 9.99 | 7.62 | 8.08 |
| 13 | 3.37 | 9.99 | 5.04 | 6.77 |
| 14 | 2.92 | 9.97 | 3.91 | 6.27 |
| 15 | 2.60 | 9.95 | 3.23 | 4.65 |
| 16 | 2.35 | 9.99 | 2.75 | 3.75 |
| 17 | 2.14 | 10.00 | 2.39 | 3.16 |
| 18 | 1.97 | 9.98 | 2.13 | 2.74 |
| 19 | 1.83 | 9.98 | 1.91 | 2.42 |
| 20 | 1.70 | 9.93 | 1.74 | 2.18 |
| $\overline{Bw}_1 = \frac{1}{10} \sum_{i=1}^{10} Bw_i$ | 7.24 | 0.02 | 6.12 | 5.14 |
| $\overline{Bw}_2 = \frac{1}{10} \sum_{i=11}^{20} Bw_i$ | 2.75 | 9.98 | 3.88 | 4.86 |
| $\overline{Bw}_3 = \frac{1}{20} \sum_{i=1}^{20} Bw_i$ | 5.00 | 5.00 | 5.00 | 5.00 |
| $Dev_1 = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (Bw_i - 5)^2}$ | 3.05 | 4.98 | 1.87 | 1.31 |
| $Dev_2 = \sqrt{\frac{1}{10} \sum_{i=11}^{20} (Bw_i - 5)^2}$ | 2.44 | 4.98 | 2.47 | 2.28 |
| $Dev_3 = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (Bw_i - 5)^2}$ | 2.76 | 4.98 | 2.19 | 1.86 |



รูปที่ 5.11 แบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 200%



รูปที่ 5.12 เส้นแนวโน้มของแบนด์วิดท์ที่โหนดได้รับ เมื่อโหลดเท่ากับ 200%

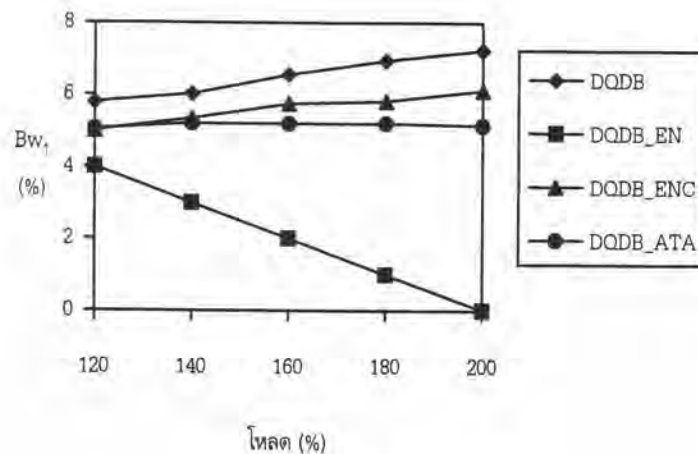
กราฟในรูปที่ 5.11 จะใกล้เคียงกับกราฟในรูปที่ 5.9 และจะเห็นความแตกต่างระหว่างโปรโตคอลโปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์ที่ใช้ตัวนับกับวิธีการปรับค่าเทรซโฮลด์ที่โหนด

อีเรเซอร์แบบปรับตัวได้ชัดเจนกว่า และจะเห็นว่าในโปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์แบบ
มูลฐานนั้นโนดที่อยู่ส่วนต้นของบัสจะได้รับแบนด์วิดท์ต่ำมาก เนื่องจากโนดที่อยู่ส่วนปลายของบัส
จะสามารถใช้สล็อตที่ถูกลบโดยโนดอีเรเซอร์ได้ และทำให้โนดเหล่านี้ทำการจองสล็อตบนบัส B
มากขึ้นด้วย ซึ่งทำให้โนดที่อยู่บริเวณต้นบัสต้องปล่อยสล็อตว่างให้ผ่านไปให้โนดที่อยู่ส่วนปลาย
ของบัสโดยที่มันไม่สามารถจะใช้สล็อตว่างเหล่านั้นได้ จึงทำให้โนดที่อยู่บริเวณต้นบัสส่งข้อมูลได้
น้อยมาก

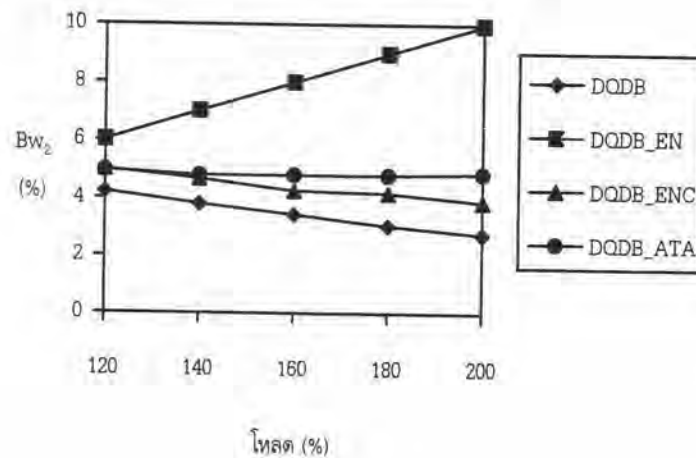
5.3 วิเคราะห์ผลการจำลองแบบ

จากผลการจำลองแบบ ในสภาวะไหลลดต่ำ (0-100%) ดังแสดงในรูปที่ 5.1 จะเห็นได้ว่า ทุกวิธีจะให้ผลลัพธ์เหมือนกัน โดยทุกโนดจะได้รับแบนด์วิดท์เท่า ๆ กัน ซึ่งแบนด์วิดท์ที่แต่ละโนด ได้รับมีค่าเท่ากับ $(100\%)/(20 \text{ โนด})$ หรือคิดเป็น 5% ของแบนด์วิดท์ทั้งหมด นั่นแสดงว่า การใช้ งานโหนดอีเธอร์เน็ตไม่ส่งผลกระทบต่อการจัดสรรแบนด์วิดท์ในโครงข่ายที่สภาวะไหลลดต่ำ

ในสภาวะไหลเกิน (มากกว่า 100 %) เราจะนำค่า \overline{Bw}_1 , \overline{Bw}_2 , Dev_1 , Dev_2 และ Dev_3 ที่ได้จากการจำลองแบบจากตารางที่ 5.2 ถึง 5.6 มาเขียนกราฟเปรียบเทียบกันได้ดังนี้

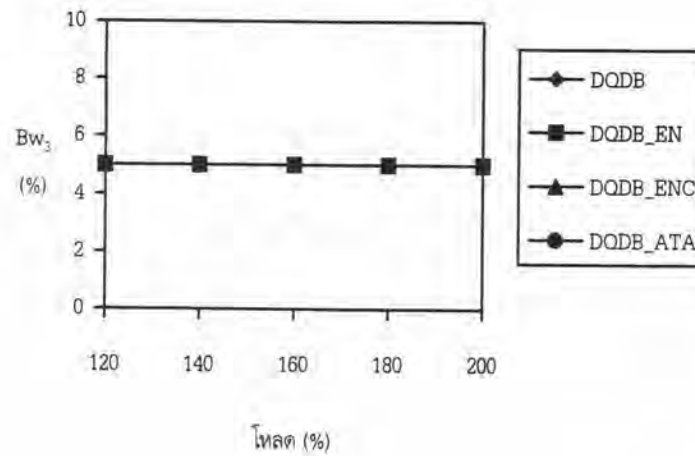


รูปที่ 5.13 ค่าแบนด์วิดท์เฉลี่ยของโนดที่ 1-10 (\overline{Bw}_1)



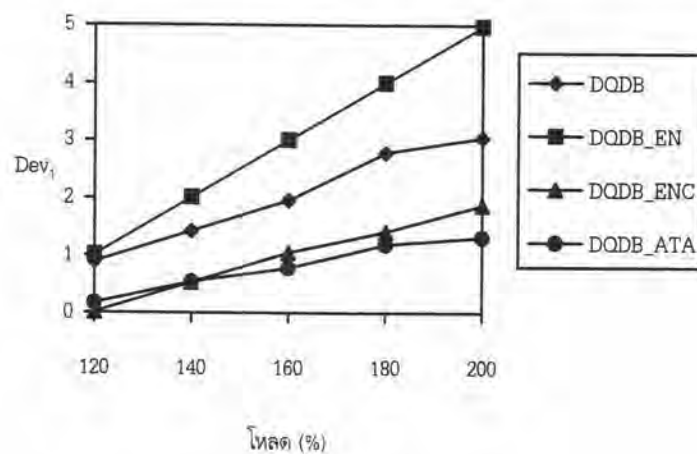
รูปที่ 5.14 ค่าแบนด์วิดท์เฉลี่ยของโหนดที่ 11-20 ($\overline{Bw_2}$)

จากรูปที่ 5.13 และ 5.14 จะเห็นได้ว่า ในขณะที่โหลตมีค่าสูงขึ้น โปรโตคอล DQDB ที่ไม่มีการใช้โหนดอีเรเซอร์ และโปรโตคอล DQDB ที่ใช้วิธีโหนดอีเรเซอร์ที่ใช้ตัวนับนั้น โหนดที่อยู่ส่วนต้นของบัล (โหนดที่ 1-10) จะได้รับแบนด์วิดท์เพิ่มขึ้น ส่วนโหนดที่อยู่ส่วนปลายของบัล (โหนดที่ 11-20) จะได้รับแบนด์วิดท์น้อยลง ซึ่งเป็นเพราะว่า โหนดที่อยู่ส่วนต้นของบัลจะสามารถใช้สล็อตว่างได้ก่อน ทำให้เหลือสล็อตว่างที่จะผ่านไปยังส่วนปลายของบัลน้อยลง ส่วนโปรโตคอล DQDB ที่ใช้วิธีโหนดอีเรเซอร์แบบมูลฐานที่ไม่มีการยกเลิกการจองสล็อตนั้น โหนดที่อยู่ส่วนปลายของบัลจะได้รับแบนด์วิดท์สูงกว่าโหนดที่อยู่ส่วนต้นของบัลมาก ทั้งนี้เป็นเพราะว่า โหนดที่อยู่ส่วนปลายของบัลจะได้รับแบนด์วิดท์ที่เพิ่มขึ้นจากสล็อตที่ถูกลบโดยโหนดอีเรเซอร์ ทำให้สามารถส่งข้อมูลได้มากขึ้น และในขณะเดียวกันก็จะทำให้เกิดการจองสล็อตมากขึ้นตามไปด้วย และเมื่อโหนดที่อยู่ส่วนต้นของบัลได้รับสัญญาณการจองสล็อตมากขึ้น มันก็จะปล่อยสล็อตว่างให้ผ่านไปมากขึ้น จึงทำให้โหนดเหล่านี้ส่งข้อมูลได้น้อยลง ส่วนโปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรซโสลด์ที่โหนดอีเรเซอร์แบบปรับตัว โหนดที่อยู่ส่วนต้นและปลายของบัลจะได้รับแบนด์วิดท์ที่ใกล้เคียงกัน



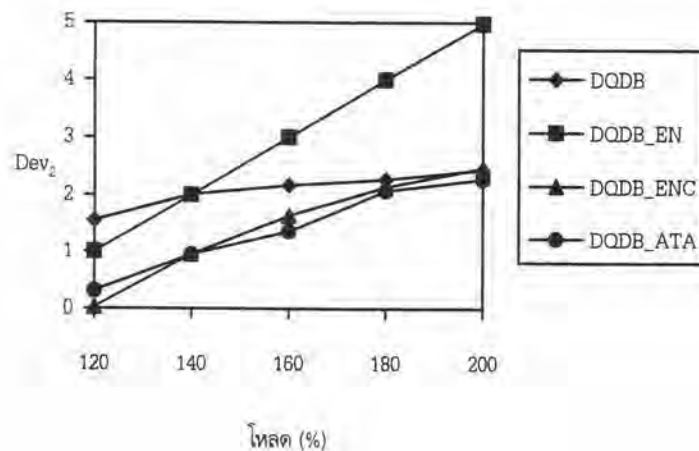
รูปที่ 5.15 ค่าแบนด์วิดท์เฉลี่ยของทั้งโครงข่าย (\overline{Bw}_3)

จากกราฟรูปที่ 5.15 จะเห็นว่า ทุกวิธีจะมีค่าเฉลี่ยของแบนด์วิดท์เท่ากันคือ 5% แต่จากผลการจำลองแบบที่ผ่านมามีทำให้เราทราบว่า ที่สภาวะโหนดเกิน แบนด์วิดท์ที่โหนดต่าง ๆ จะได้รับนั้นมีค่าแตกต่างกัน ดังนั้น เราจึงจำเป็นต้องพิจารณาค่าเบี่ยงเบนของแบนด์วิดท์ประกอบด้วย



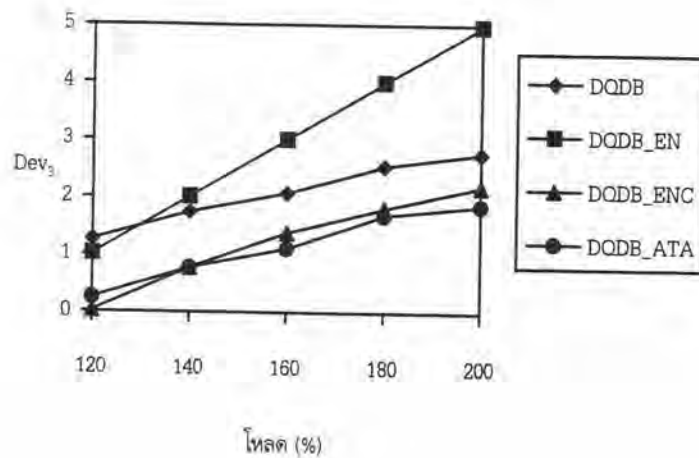
รูปที่ 5.16 ส่วนเบี่ยงเบนของแบนด์วิดท์ของโหนดที่ 1-10 (Dev_1)

เมื่อพิจารณาค่า Dev_1 หรือค่าส่วนเบี่ยงเบนของแบนด์วิดท์ที่ได้รับของโนดที่อยู่บริเวณ ส่วนต้นของบัส (โนดที่ 1 ถึง 10) ดังแสดงในรูปที่ 5.16 จะเห็นว่า ทุกวิธีจะมีค่า Dev_1 สูงขึ้นเมื่อ โหลดมีค่าเพิ่มขึ้น โดยโปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์แบบมูลฐานที่ไม่มีการยกเลิกการจอง สล็อตจะมีค่า Dev_1 สูงที่สุด รองลงมาได้แก่โปรโตคอล DQDB ที่ไม่มีการใช้โนดอีเรเซอร์ และโปร โตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์ที่ใช้ตัวนับจะมีค่าต่ำที่สุดในช่วงที่โหลดมีค่าระหว่าง 120% ถึง 140% แต่เมื่อโหลดมีค่ามากกว่า 140% แล้ว โปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรชโฮลด์ที่โนด อีเรเซอร์แบบปรับตัวจะมีค่าต่ำที่สุด นั่นแสดงว่า โปรโตคอล DQDB ที่ใช้วิธีการปรับค่าเทรชโฮลด์ ที่โนดอีเรเซอร์แบบปรับตัวจะทำให้แบนด์วิดท์ของโนดที่อยู่บริเวณส่วนต้นของบัสมีความแตกต่าง กันน้อยที่สุด หรือกล่าวได้ว่ามีการจัดสรรแบนด์วิดท์ได้ดีกว่าเมื่อเทียบกับวิธีอื่น ๆ



รูปที่ 5.17 ส่วนเบี่ยงเบนของแบนด์วิดท์ของโนดที่ 11-20 (Dev_2)

จากรูปที่ 5.17 ซึ่งแสดงความสัมพันธ์ระหว่างค่าส่วนเบี่ยงเบนของแบนด์วิดท์ของโนดที่ 11-20 (Dev_2) กับค่าโหลด จะเห็นว่า ที่โหลดมากกว่า 180% ค่า Dev_2 ของวิธีที่ไม่มีการใช้โนด อีเรเซอร์, วิธีโนดอีเรเซอร์ที่ใช้ตัวนับ และวิธีการปรับค่าเทรชโฮลด์ที่โนดอีเรเซอร์แบบปรับตัวจะมีค่า ใกล้เคียงกัน เพราะแบนด์วิดท์ส่วนใหญ่จะอยู่ที่บริเวณโนดอีเรเซอร์และจะมีค่าลดลงเรื่อย ๆ เมื่อ เข้าใกล้ปลายบัส ดังแสดงในรูปที่ 5.9 และ 5.11 จึงทำให้การกระจายของแบนด์วิดท์ไม่แตกต่าง กันมากนัก



รูปที่ 5.18 ส่วนเบี่ยงเบนของแบนด์วิดท์ของทั้งโครงข่าย (Dev_3)

ในรูปที่ 5.18 จะแสดงค่าของ Dev_3 หรือค่าส่วนเบี่ยงเบนของแบนด์วิดท์ที่โหนดได้รับของทั้งโครงข่าย จะเห็นว่า โดยภาพรวมแล้วโปรโตคอล DQDB ที่ใช้วิธีโนดอีเรเซอร์แบบมูลฐานที่ไม่มี การยกเลิกการจองสล็อตจะมีการจัดสรรแบนด์วิดท์ไปยังโหนดต่าง ๆ ได้ไม่ดี เพราะแบนด์วิดท์ที่ได้รับจะแตกต่างกันมาก และเมื่อโหลดมีค่าสูงเกิน 140% ไปแล้วโปรโตคอล DQDB ที่ใช้วิธีการปรับ ค่าเทรซโบลด์ที่โนดอีเรเซอร์แบบปรับตัวจะมีการจัดสรรแบนด์วิดท์ไปยังโหนดต่าง ๆ ได้ดีที่สุดในเมื่อเทียบกับวิธีอื่น ๆ