

ผลการจำลองและการวิเคราะห์

บทนี้จะทำการจำลองการทำงานของอัลกอริทึมที่ได้เสนอขึ้นมาเพื่อปรับปรุงให้โครงข่ายมีประสิทธิภาพมากขึ้น โดยจะกล่าวถึงผลการจำลองการทำงานของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตเพื่อปรับปรุงสมรรถนะของ TCP โดยเปรียบเทียบกับอัลกอริทึม EPD และ อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting บนแบบจำลอง Peer to Peer, Parking Lot และ Chain โดยมีแหล่งกำเนิดเป็นแบบส่งข้อมูลตลอดเวลาและแบบเบิรสต์ ซึ่งข้อกำหนดต่างๆของแบบจำลองได้กล่าวมาแล้วในบทที่ 4 ในส่วนสุดท้ายจะเป็นการปรับพารามิเตอร์ของ TCP เพื่อศึกษาผลกระทบพารามิเตอร์ของ TCP กับอัลกอริทึมแบบต่างๆ

5.1 ผลการจำลองการทำงานของแบบจำลอง Peer to Peer

5.1.1 ผลการจำลองการทำงานโดยที่แหล่งกำเนิดเป็นแบบส่งตลอดเวลา (persistent) ทั้งหมด ในโครงข่าย Local Area Network (LAN) และ Wide Area Network (WAN)

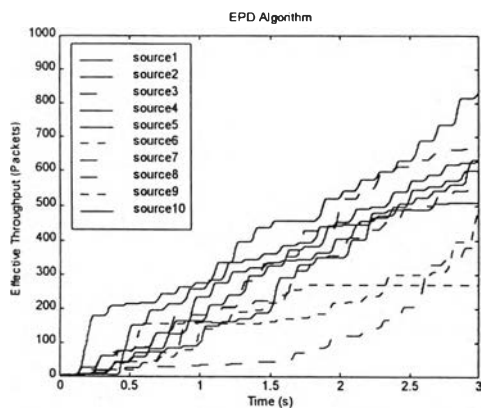
ในส่วนนี้จะทำการจำลองการทำงานกับอัลกอริทึมทั้ง 3 คือ อัลกอริทึม EPD, อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting และอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตใน โครงข่ายแบบ LAN และ WAN การจำลองในส่วนนี้ทดสอบกับแหล่งกำเนิดที่มีการส่งข้อมูลตลอดเวลาเป็นจำนวน 10 แหล่งกำเนิดและขนาดบัฟเฟอร์ในสวิตช์มีขนาดจำกัด โดยในการจำลองนี้ให้ขนาดบัฟเฟอร์มีค่า 2000 เซลล์ ระดับจุดเริ่มเปลี่ยนของอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting มีค่าเป็น 1800 เซลล์ และระดับจุดเริ่มเปลี่ยนด้านสูง (High Threshold) และด้านต่ำ (Low Threshold) ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนแบบพลวัตมีค่าเป็น 1800 และ 1000 เซลล์ ตามลำดับ สาเหตุที่เลือกใช้จุดเริ่มเปลี่ยนด้านต่ำและสูงดังที่กล่าวมาเพราะจะทำการศึกษาถึงปัญหาการเกิดแพ็กเก็ตไม่สมบูรณ์ในโครงข่าย ส่วนข้อกำหนดอื่นๆของแบบจำลอง Peer to Peer เป็นไปตามส่วนที่ 4.1.1

ในตารางที่ 5.1 แสดงผลการจำลองของอัลกอริทึมทั้ง 3 โดยที่แหล่งกำเนิดมีการส่งข้อมูลตลอดเวลาในโครงข่ายแบบ LAN ตารางที่ 5.2 แสดงผลการทำงานของอัลกอริทึมทั้ง 3 โดยที่แหล่งกำเนิดมีการส่งข้อมูลตลอดเวลา ในโครงข่ายแบบ WAN รูปที่ 5.1 ถึง 5.3 แสดง Effective Throughput ของอัลกอริทึมทั้ง 3 ที่มีแหล่งกำเนิดส่งข้อมูลตลอดเวลาในโครงข่ายแบบ LAN รูปที่ 5.4 ถึง 5.6 แสดง Effective Throughput ของอัลกอริทึมทั้ง 3 ที่มีแหล่งกำเนิดส่งข้อมูลตลอดเวลาในโครงข่ายแบบ WAN

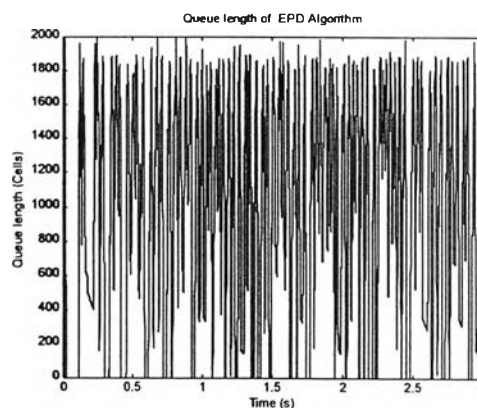
❖ ผลการจำลองของแบบจำลอง Peer to Peer ในโครงข่ายแบบ LAN

ตารางที่ 5.1 แสดงผลการจำลองของแหล่งกำเนิดเป็นแบบส่งข้อมูลตลอดเวลาในโครงข่ายแบบ LAN

| Algorithm | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packet) | 5564 | 5463 | 5900 |
| TCP Throughput (Mbit/s) | 117.96 | 115.45 | 125.39 |
| Fairness index | 0.9326 | 0.9505 | 0.9673 |
| Efficiency (%) | 87.14 | 85.28 | 92.63 |



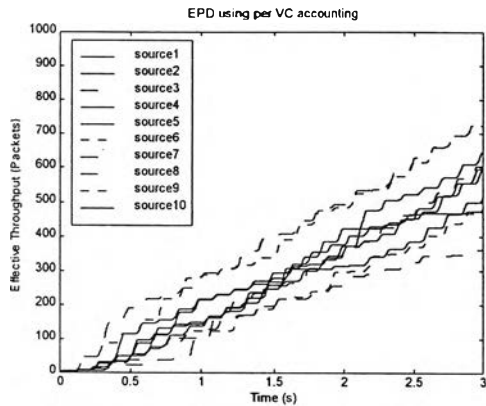
(ก)



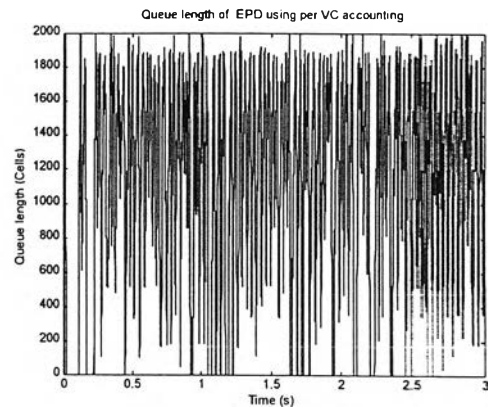
(ข)

รูปที่ 5.1 (ก) แสดง Effective Throughput ของอัลกอริทึม Early Packet Discard ในโครงข่าย LAN

(ข) ความยาวคิว ของอัลกอริทึม Early Packet Discard ในโครงข่าย LAN



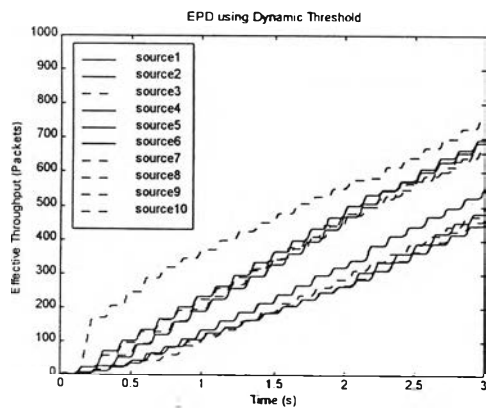
(ก)



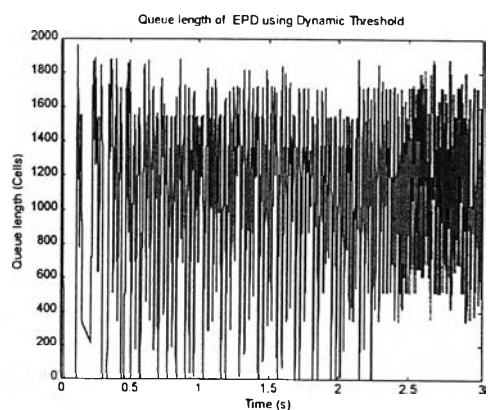
(ข)

รูปที่ 5.2 (ก) แสดง Effective Throughput ของอัลกอริทึม EPD ที่ใช้ per-VC accounting ในโครงข่าย LAN

(ข) แสดงความยาวคิว ของอัลกอริทึม EPD ที่ใช้ per-VC accounting ในโครงข่าย LAN



(ก)



(ข)

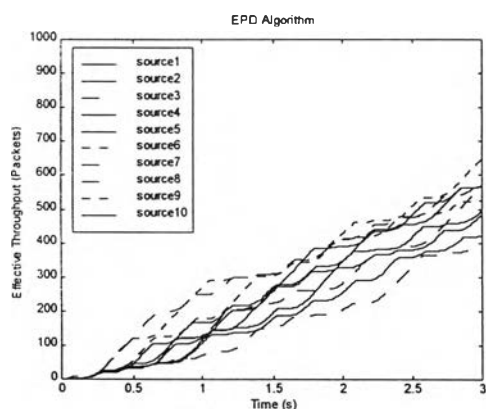
รูปที่ 5.3 (ก) แสดง Effective Throughput ของอัลกอริทึม EPD ที่ใช้จุดเริ่มเปลี่ยนพลวัต ในโครงข่าย LAN

(ข) แสดงความยาวคิว ของอัลกอริทึม EPD ที่ใช้จุดเริ่มเปลี่ยนพลวัต ในโครงข่าย LAN

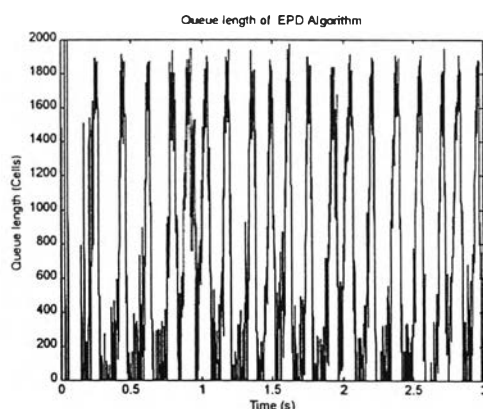
❖ ผลการจำลองของแบบจำลอง Peer to Peer ในโครงข่ายแบบ WAN

ตารางที่ 5.2 แสดงผลการจำลองของแหล่งกำเนิดเป็นแบบส่งข้อมูลตลอดเวลาในโครงข่ายแบบ WAN

| Algorithm | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packet) | 5244 | 4943 | 4968 |
| TCP Throughput (Mbit/s) | 110.96 | 104.6 | 104.85 |
| Fairness index | 0.9810 | 0.9822 | 0.9906 |
| Efficiency (%) | 81.97 | 77.27 | 77.45 |



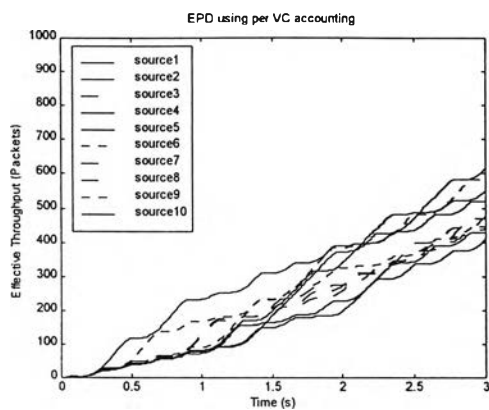
(ก)



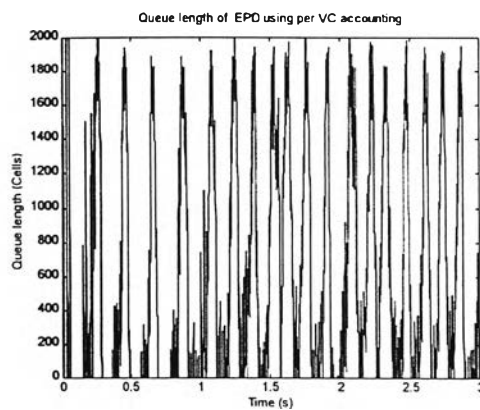
(ข)

รูปที่ 5.4 (ก) แสดง Effective Throughput ของอัลกอริทึม Early Packet Discard ในโครงข่าย WAN

(ข) ความยาวคิว ของอัลกอริทึม Early Packet Discard ในโครงข่าย WAN

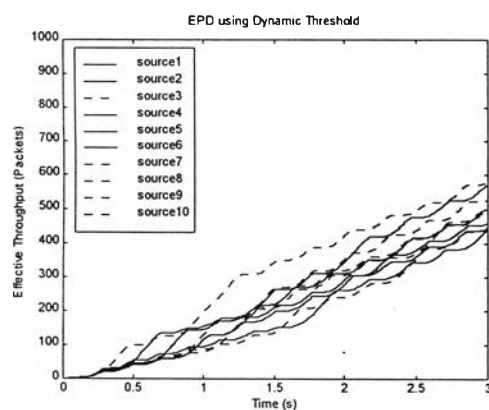


(ก)

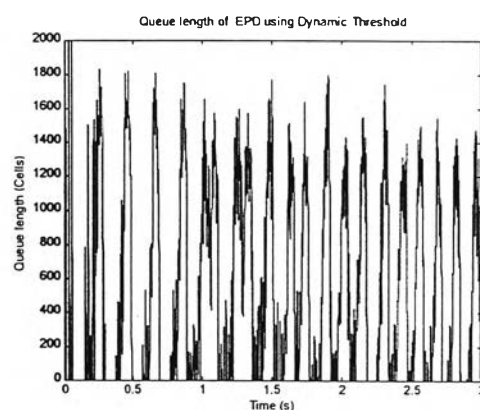


(ข)

รูปที่ 5.5 (ก) แสดง Effective Throughput ของอัลกอริทึม EPD ที่ใช้ per-VC accounting ในโครงข่าย WAN
 (ข) แสดงความยาวคิว ของอัลกอริทึม EPD ที่ใช้ per-VC accounting ในโครงข่าย WAN



(ก)



(ข)

รูปที่ 5.6 (ก) แสดง Effective Throughput ของอัลกอริทึม EPD ที่ใช้จุดเริ่มเปลี่ยนพลวัต ในโครงข่าย WAN
 (ข) แสดงความยาวคิว ของอัลกอริทึม EPD ที่ใช้จุดเริ่มเปลี่ยนพลวัต ในโครงข่าย WAN

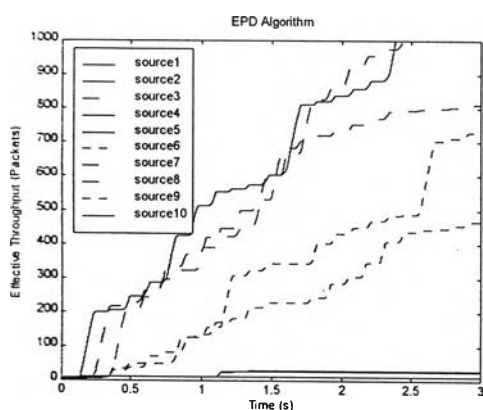
5.1.2 ผลการจำลองการทำงานโดยที่แหล่งกำเนิดเป็นแบบส่งต่อเนื่อง (persistent) และแบบเบิรสต์ (burst) ในโครงข่าย Local Area Network (LAN)

แหล่งกำเนิดที่มีการส่งข้อมูลเป็นแบบเบิรสต์ คือแหล่งกำเนิดที่มีการส่งข้อมูลเป็นช่วงๆ บางช่วงมีการส่งข้อมูลและบางช่วงไม่มีการส่งข้อมูล โดยในการจำลองการทำงานในส่วนนี้จะมีแหล่งกำเนิดที่เป็นแบบส่งต่อเนื่องและแบบส่งเป็นช่วงๆ ซึ่งจะเป็นการทดสอบถึงการปรับตัวในการส่งข้อมูลของแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่องเมื่อมีแหล่งกำเนิดที่ส่งข้อมูลเป็นเบิรสต์ แบบจำลองที่ใช้ในการทดสอบนี้เป็นแบบจำลองในรูปที่ 4.1 โดยมีข้อกำหนดของแบบจำลองดังในหัวข้อที่ 4.1.1 จากรูปที่ 4.1 แหล่งกำเนิดที่ 1 ถึง 5 เป็นแหล่งกำเนิดแบบเบิรสต์ และแหล่งกำเนิดที่ 6 ถึง 10 เป็นแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่อง โดยช่วงเวลาที่มีการส่งข้อมูลของแหล่งกำเนิดแบบเบิรสต์จะส่งข้อมูล 100 KB และช่วงเวลาไม่ส่งข้อมูลของแหล่งกำเนิดแบบเบิรสต์จะมีค่าเป็น 1 วินาที ซึ่งแหล่งกำเนิดทั้งสองแบบนี้จะถูกควบคุมการส่งข้อมูลเข้าไปในโครงข่ายโดย TCP อีกทีหนึ่ง TCP จะใช้ window เป็นกลไกในการควบคุมการส่งข้อมูล

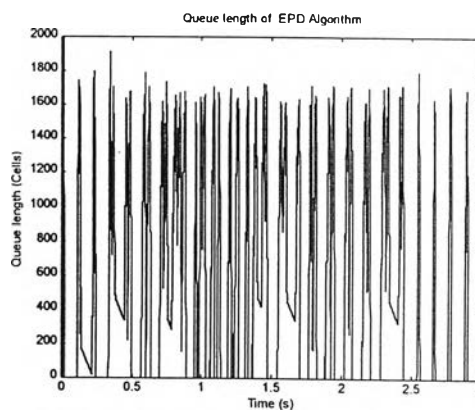
ตารางที่ 5.3 แสดงผลการจำลองการทำงานของอัลกอริทึม EPD ทั้ง 3 โดยมีแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่องและส่งข้อมูลเป็นเบิรสต์ ในโครงข่ายแบบ LAN รูปที่ 5.7 ถึง 5.9 แสดง Effective Throughput ของอัลกอริทึมทั้ง 3 ที่มีแหล่งกำเนิดส่งข้อมูลต่อเนื่องและส่งข้อมูลแบบเบิรสต์ ในโครงข่ายแบบ LAN

ตารางที่ 5.3 แสดงผลการจำลองของแบบจำลอง Peer to Peer โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง และแบบเบิร์ตส์ ในโครงข่ายแบบ LAN

| Algorithm | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packet) | 4507 | 5037 | 5046 |
| TCP Throughput (Mbit/s) | 92.989 | 103.02 | 104.7 |
| Fairness index | 0.8970 | 0.9547 | 0.9806 |
| Efficiency (%) | 68.69 | 76.10 | 77.35 |

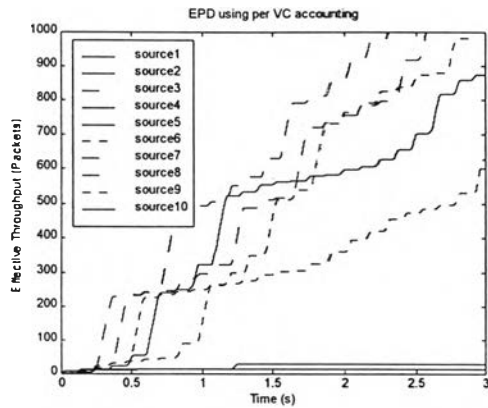


(ก)

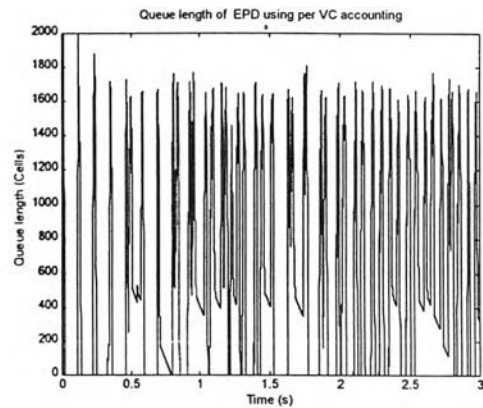


(ข)

รูปที่ 5.7 (ก) แสดง Effective Throughput (ข) ความยาวคิว ของอัลกอริทึม Early Packet Discard โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิร์ตส์ ในโครงข่าย LAN

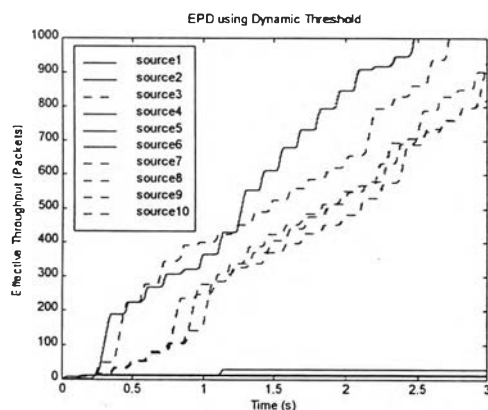


(ก)

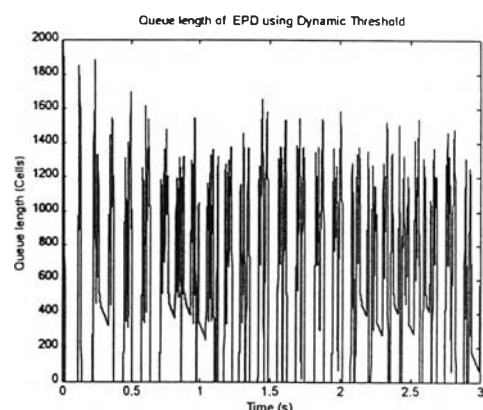


(ข)

รูปที่ 5.8 (ก) แสดง Effective Throughput (ข) ความยาวคิว ของอัลกอริทึม EPD ที่ใช้ per-VC accounting โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิรสต์ ใน โครงข่าย LAN



(ก)



(ข)

รูปที่ 5.9 (ก) แสดง Effective Throughput (ข) ความยาวคิว ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยน พลวัตโดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิรสต์ ใน โครงข่าย LAN

5.1.3 การวิเคราะห์ผลของแบบจำลอง Peer to Peer

จากผลการจำลองการทำงานของแบบจำลอง Peer to Peer โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องในโครงข่ายแบบ LAN อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting จะมีค่าวิสัยสามารถ (Throughput) ต่ำสุดจึงเป็นสาเหตุให้ประสิทธิภาพของ TCP ลดต่ำลงด้วย เนื่องจากอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting เกิดการล้นในบัฟเฟอร์ของสวิตช์มากที่สุด ดังในรูปที่ 5.2 (ข) จึงทำให้มีบางส่วนของแพ็กเก็ตถูกส่งเข้าไปในโครงข่าย เพราะเมื่อเกิดมีบางส่วนของแพ็กเก็ตถูกส่งเข้าไปในโครงข่ายจะทำให้มีการสูญเสียแบนด์วิดท์ โดยบางส่วนของแพ็กเก็ตนี้เมื่อถึงปลายทางแล้ว ปลายทางไม่สามารถประกอบเป็นแพ็กเก็ตที่สมบูรณ์ได้ ดังนั้นปลายทางจึงส่ง ACK กลับไปบอกให้แหล่งกำเนิดส่งแพ็กเก็ตที่สูญเสียนั้นมาใหม่ บางส่วนของแพ็กเก็ตที่ถูกส่งเข้าไปในโครงข่ายจะทำให้มีการสูญเสียแบนด์วิดท์แทนที่จะนำแบนด์วิดท์ส่วนนี้ไปให้กับแพ็กเก็ตอื่นๆ จากตารางที่ 5.1 ประสิทธิภาพ TCP ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตจะมีความมากที่สุดเพราะมีจำนวนบางส่วนของแพ็กเก็ตที่ส่งเข้าไปในโครงข่ายน้อยที่สุด และจุดเริ่มเปลี่ยนพลวัตจะทำให้แบนด์วิดท์ที่ยังเหลืออยู่แบ่งไปให้กับ VC ที่ยังต้องการส่งข้อมูลอีก ดังในรูปที่ 5.3 (ข) อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตสามารถปรับปรุงประสิทธิภาพ TCP เพิ่มขึ้นจากอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ประมาณ 5% และ 7% ตามลำดับ

ในส่วนความเท่าเทียมกัน (Fairness) อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting และที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตจะมีการนับจำนวนเซลล์ที่เข้ามาในสวิตช์ โดยจะทำการทิ้งเซลล์ออกไปเมื่อ VC ใดๆมีจำนวนเซลล์ที่อยู่ในบัฟเฟอร์มากเกินกว่าระดับ fair share (ระดับ fair share คือระดับเฉลี่ยของแต่ละ VC จะมีจำนวนเซลล์อยู่ในบัฟเฟอร์) จึงทำให้อัลกอริทึมทั้งสองนี้สามารถปรับปรุงความเท่าเทียมกันได้ ส่วนอัลกอริทึม EPD จะไม่มีการนับจำนวนเซลล์ใดๆและไม่สนใจว่าจำนวนเซลล์ของ VC ใดๆจะมีมากเกินกว่าระดับ fair share หรือไม่ อัลกอริทึม EPD จะสนใจอยู่เพียงว่าความยาวคิวขณะที่มีเซลล์แรกของแพ็กเก็ตเข้ามาใหม่มากเกินกว่าจุดเริ่มเปลี่ยนแล้วหรือไม่ ถ้ามากเกินกว่าจุดเริ่มเปลี่ยนสวิตช์ก็จะทำการทิ้งเซลล์ออกไปโดยไม่คำนึงว่าแพ็กเก็ตที่เข้ามาใหม่ของ VC นั้นจะมีจำนวนแพ็กเก็ตที่อยู่ในบัฟเฟอร์มากกว่าระดับ fair share หรือไม่ ฉะนั้นอัลกอริทึม EPD จะทำการทิ้งเซลล์เป็นแบบสุ่ม (random) จึงเป็นสาเหตุให้ค่าความเท่าเทียมกันลดลงกว่าอัลกอริทึมอื่น

ในโครงข่ายแบบ WAN ซึ่งโครงข่ายแบบ WAN เป็นโครงข่ายที่มีระยะทางไกล จากผลการจำลองการทำงานโดยแหล่งกำเนิดเป็นแบบส่งข้อมูลอย่างต่อเนื่องในโครงข่ายแบบ WAN จะพบว่าประสิทธิภาพ TCP ของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting และเทคนิคจุดเริ่มเปลี่ยนพลวัตมีค่าใกล้เคียงกัน เนื่องจากทั้งสองอัลกอริทึมทำการทิ้งเซลล์ออกเป็นจำนวนมากเพื่อทำให้แต่ละแหล่งกำเนิดส่งข้อมูลได้เท่าเทียมกัน โดยแหล่งกำเนิดไม่มีจำนวนเซลล์ในบัฟเฟอร์มากเกินกว่าระดับ fair share เมื่อมีแพ็กเก็ตใหม่เข้ามาที่บัฟเฟอร์ สวิตช์จะทิ้งแพ็กเก็ตที่เข้ามาใหม่นั้นออกไปเพื่อปรับให้แต่ละแหล่งกำเนิดมีความเท่าเทียมกันใน

การส่งข้อมูล ฉะนั้นเมื่อสวิตช์มีการทิ้งเซลล์ออกเป็นจำนวนมากทำให้ค่าวิสัยสามารถและประสิทธิภาพ TCP มีค่าลดลงประมาณ 4% จากอัลกอริทึม EPD แต่การทิ้งเซลล์ของแพ็กเก็ตที่เข้ามาใหม่ออกไปเมื่อจำนวนเซลล์ของ VC นั้นมีจำนวนมากเกินกว่าระดับ fair share จะทำให้ค่าความเท่าเทียมกันเข้าใกล้ 1 มากกว่า (ค่าความเท่าเทียมกันเท่ากับ 1 หมายถึงทุกๆแหล่งกำเนิดส่งข้อมูลได้เท่าเทียมกันหรือเสมอภาคกัน) และทำให้จำนวนแพ็กเก็ตที่รับได้หรือ Effective Throughput น้อยกว่าอัลกอริทึม EPD

สำหรับผลการจำลองการทำงานที่มีแหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องตลอดเวลาและแบบเบิรสต์ จะพบว่าประสิทธิภาพของ TCP ลดต่ำกว่าผลการจำลองที่มีแหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องตลอดเวลา เนื่องจากบางช่วงเวลาของแหล่งกำเนิดแบบเบิรสต์หยุดส่งข้อมูล ทำให้มีช่วงเวลาที่ไม่มีข้อมูลส่งเข้ามาที่โครงข่าย และอีกสาเหตุหนึ่งที่ทำให้ประสิทธิภาพของ TCP ลดลงเพราะสวิตช์ทำการทิ้งเซลล์ของแพ็กเก็ตออกจากสวิตช์พร้อมๆกันหลายๆแหล่งกำเนิดและในขณะนั้นแหล่งกำเนิดแบบเบิรสต์อยู่ในช่วงเวลาที่ไม่ส่งข้อมูล จึงทำให้ความยาวคิวในบัฟเฟอร์ของสวิตช์ลดต่ำลงอย่างรวดเร็ว ดังแสดงในรูปที่ 5.7(ก), 5.8 (ก), และ 5.9(ก)

แหล่งกำเนิดที่เป็นแบบเบิรสต์ในช่วงเวลาที่ส่งข้อมูลจะส่งข้อมูล 100 Kbytes และเมื่อส่งข้อมูลครบแล้วก็จะหยุดส่งข้อมูลเป็นเวลา 1 วินาที โดยข้อมูล 100 Kbytes เมื่อส่งเข้าไปในชั้น TCP จะแบ่งเป็นแพ็กเก็ตได้ทั้งหมด 13 แพ็กเก็ต จากรูปที่ 5.7(ก), 5.8(ก) และ 5.9(ก) จะมีแหล่งกำเนิดแบบเบิรสต์ที่สามารถส่งข้อมูลได้เพียง 1 ครั้งหรือเพียง 13 แพ็กเก็ตเท่านั้นและบางแหล่งกำเนิดแบบเบิรสต์สามารถส่งข้อมูลได้ 2 ครั้ง สาเหตุที่แหล่งกำเนิดแบบเบิรสต์ไม่สามารถส่งข้อมูลได้เท่ากันเพราะในช่วงเวลาที่ส่งข้อมูลของแหล่งกำเนิดแบบเบิรสต์ ความยาวคิวในบัฟเฟอร์ของสวิตช์มากเกินกว่าจุดเริ่มเปลี่ยน EPD หรือจุดเริ่มเปลี่ยนด้านสูง (อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต) ทำให้แพ็กเก็ตของแหล่งกำเนิดแบบเบิรสต์ถูกทิ้งออกไปที่สวิตช์ ส่วนแหล่งกำเนิดแบบเบิรสต์ที่สามารถส่งข้อมูลได้มากกว่า 1 ครั้งเพราะแพ็กเก็ตของแหล่งกำเนิดแบบเบิรสต์ส่งเข้าไปในสวิตช์ในช่วงจังหวะเวลาที่มีเนื้อที่ว่างอยู่ในบัฟเฟอร์ของสวิตช์พอดี จึงทำให้แพ็กเก็ตของแหล่งกำเนิดแบบเบิรสต์ไม่ถูกทิ้งออกไปจากสวิตช์

เมื่อพิจารณาถึงค่าความเท่าเทียมกันของผลการจำลองที่มีแหล่งกำเนิดเป็นแบบเบิรสต์และแบบส่งข้อมูลต่อเนื่องตลอดเวลา วิทยานิพนธ์ฉบับนี้จะคิดค่าความเท่าเทียมกันเฉพาะแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่องตลอดเวลาเท่านั้น เนื่องจากในอัลกอริทึม EPD ไม่มีการนับจำนวนเซลล์ที่อยู่ในสวิตช์ทำให้ไม่ว่า VC ใดมีการใช้งานมากหรือน้อย และควรจะทิ้งแพ็กเก็ตของ VC ใดเมื่อโครงข่ายเกิดความคับคั่งทำให้อัลกอริทึม EPD ทำการทิ้งแพ็กเก็ตแบบสุ่ม จึงทำให้ค่าความเท่าเทียมกันต่ำ อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting มีการนับจำนวนเซลล์ในบัฟเฟอร์ของสวิตช์ทำให้รู้ว่า VC ใดมีการใช้งานมากหรือน้อย จึงทำให้ค่าความเท่าเทียมกันดีขึ้นกว่าอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting 5.77% ส่วนอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตเป็นอัลกอริทึมหนึ่งที่มีการนับจำนวนเซลล์ที่เข้ามาในบัฟเฟอร์ของสวิตช์

และมีการแบ่งแบนด์วิดท์ของแหล่งกำเนิดแบบเบิรสต์ขณะอยู่ในช่วงหยุดส่งข้อมูลมาให้แหล่งกำเนิดแบบส่งข้อมูลต่อเนื่องตลอดเวลา จึงทำให้อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตสามารถปรับปรุงประสิทธิภาพของ TCP และความเท่าเทียมกันในการส่งข้อมูลให้ดีขึ้นกว่าอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting โดยอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีประสิทธิภาพดีกว่าอัลกอริทึม EPD และ อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ประมาณ 8.66% และ 1.25% ตามลำดับ ส่วนค่าความเท่าเทียมกันของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมากกว่าอัลกอริทึม EPD และ อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting 8.36% และ 2.59% ตามลำดับ

5.2 ผลการจำลองการทำงานของแบบจำลอง Parking Lot

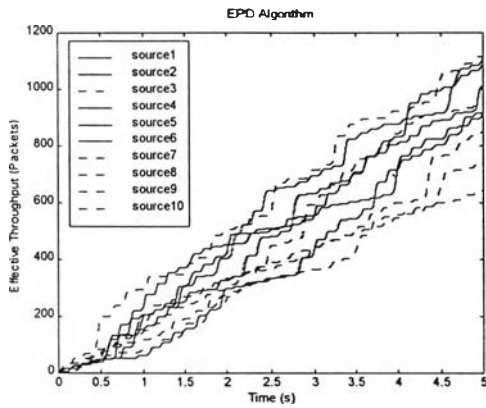
5.2.1 ผลการจำลองการทำงานโดยที่แหล่งกำเนิดเป็นแบบส่งต่อเนื่อง (persistent) ทั้งหมด ในโครงข่าย Local Area Network (LAN) และ Wide Area Network (WAN)

ในส่วนนี้จะทำการจำลองการทำงานอัลกอริทึมทั้ง 3 บนแบบจำลอง Parking Lot ที่มีจำนวนแหล่งกำเนิดทั้งหมด 10 แหล่งกำเนิดและมีการส่งข้อมูลต่อเนื่อง โดยมีแหล่งกำเนิดจำนวน 4 แหล่งกำเนิดที่มีการแทรกเข้ามาระหว่างทางดูในรูปที่ 4.2 ซึ่งแหล่งกำเนิดที่แทรกเข้ามาจะมีระยะทางไม่เท่ากันจึงทำให้มีค่าประวิงเวลาไม่เท่ากัน แบบจำลอง Parking Lot เป็นแบบจำลองหนึ่งที่ศึกษาผลของการประวิงเวลาของแต่ละแหล่งกำเนิดที่ไม่เท่ากัน โดยในการจำลองนี้ให้ขนาดบัฟเฟอร์มีค่า 2000 เซลล์ ระดับจุดเริ่มเปลี่ยนของอัลกอริทึม EPD มีค่าเป็น 1600 เซลล์ (80%ของขนาดบัฟเฟอร์) และระดับจุดเริ่มเปลี่ยนด้านสูง (High Threshold) และด้านต่ำ (Low Threshold) ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนแบบพลวัตมีค่าเป็น 1600 และ 1000 เซลล์ ตามลำดับ ส่วนข้อกำหนดอื่นๆของแบบจำลอง Parking Lot เป็นไปดังส่วนที่ 4.1.2

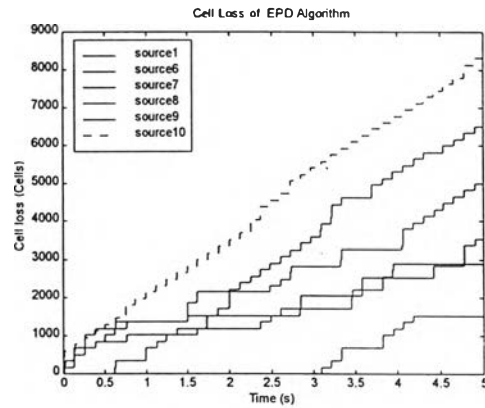
ในตารางที่ 5.4 แสดงผลการจำลองของอัลกอริทึมทั้ง 3 โดยที่แหล่งกำเนิดมีการส่งข้อมูลต่อเนื่องในโครงข่ายแบบ LAN ตารางที่ 5.5 แสดงผลการทำงานของอัลกอริทึมทั้ง 3 โดยที่แหล่งกำเนิดมีการส่งข้อมูลต่อเนื่อง ในโครงข่ายแบบ WAN รูปที่ 5.10 ถึง 5.12 แสดง Effective Throughput ของอัลกอริทึมทั้ง 3 ที่มีแหล่งกำเนิดส่งข้อมูลต่อเนื่องในโครงข่ายแบบ LAN รูปที่ 5.13 ถึง 5.15 แสดง Effective Throughput ของอัลกอริทึมทั้ง 3 ที่มีแหล่งกำเนิดส่งข้อมูลต่อเนื่องในโครงข่ายแบบ WAN

ตารางที่ 5.4 ผลการจำลองของแบบจำลอง Parking Lot โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่ายแบบ LAN

| Algorithm | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packet) | 9321 | 9854 | 10091 |
| TCP Throughput (Mbit/s) | 118.85 | 125.18 | 128.03 |
| Fairness index | 0.9745 | 0.9824 | 0.9951 |
| Efficiency (%) | 87.80 | 92.47 | 94.58 |

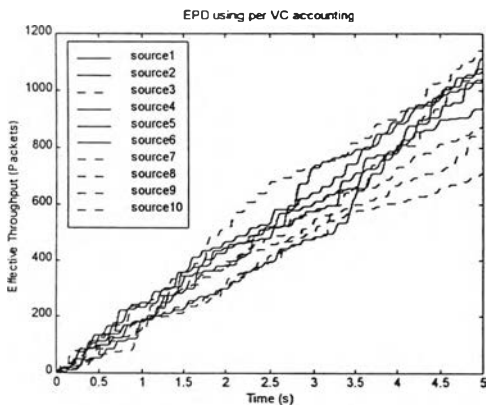


(ก)

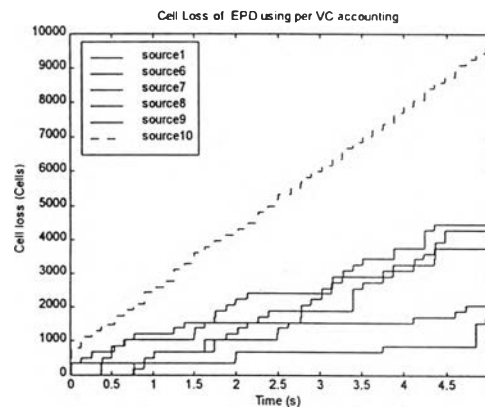


(ข)

รูปที่ 5.10 (ก) แสดง Effective Throughput (ข) Cell Loss ของอัลกอริทึม Early Packet Discard โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่าย LAN

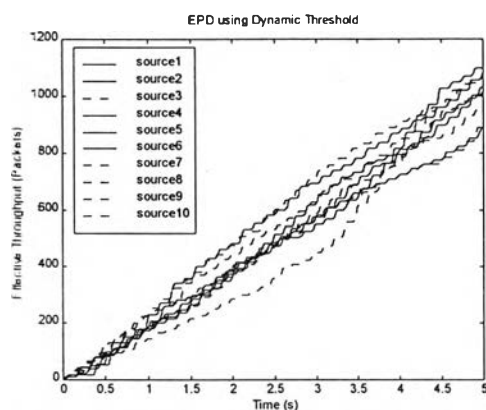


(ก)

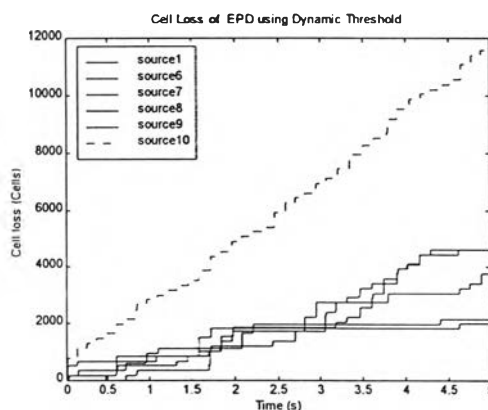


(ข)

รูปที่ 5.11 (ก) แสดง Effective Throughput (ข) Cell Loss ของอัลกอริทึม EPD ที่ใช้ per-VC accounting โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่าย LAN



(ก)

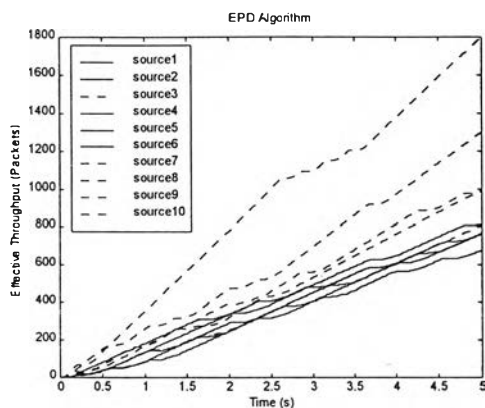


(ข)

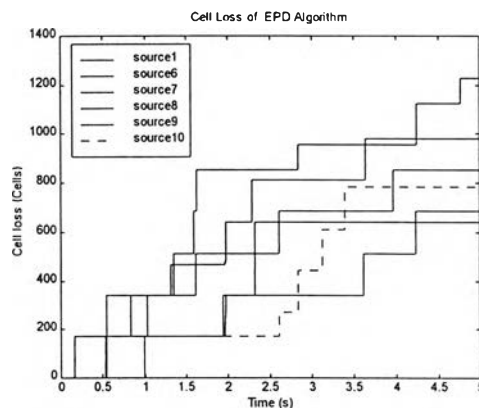
รูปที่ 5.12 (ก) แสดง Effective Throughput (ข) Cell Loss ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยน พลวัตโดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่าย LAN

ตารางที่ 5.5 ผลการจำลองของแบบจำลอง Parking Lot โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่ายแบบ WAN

| Algorithm | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packet) | 9657 | 9414 | 9208 |
| TCP Throughput (Mbit/s) | 122.28 | 119.097 | 116.67 |
| Fairness index | 0.8984 | 0.8938 | 0.8987 |
| Efficiency (%) | 90.33 | 87.98 | 86.19 |

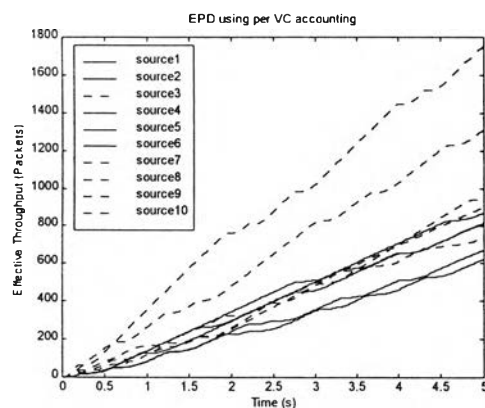


(ก)

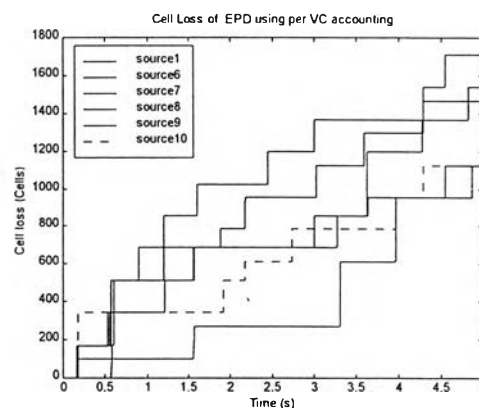


(ข)

รูปที่ 5.13 (ก) แสดง Effective Throughput (ข) Cell Loss ของอัลกอริทึม Early Packet Discard โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่าย WAN

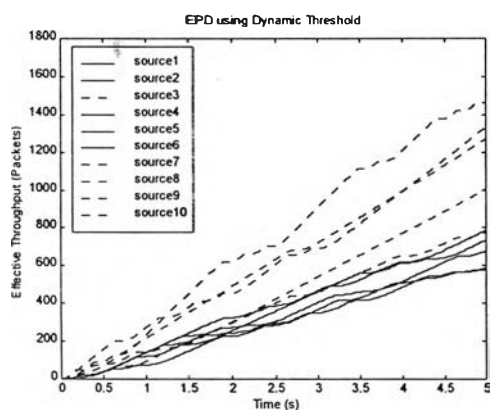


(ก)

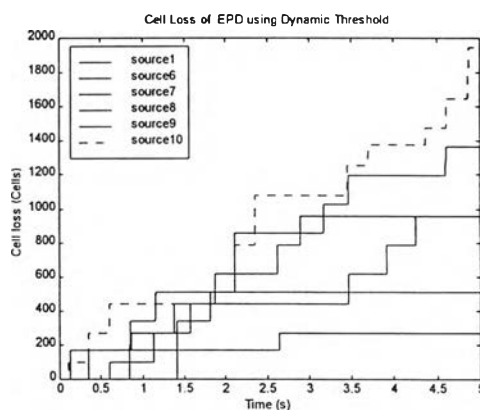


(ข)

รูปที่ 5.14 (ก) แสดง Effective Throughput (ข) Cell Loss ของอัลกอริทึม EPD ที่ใช้ per-VC accounting โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องในโครงข่าย LAN



(ก)



(ข)

รูปที่ 5.15 (ก) แสดง Effective Throughput (ข) Cell Loss ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่าย WAN

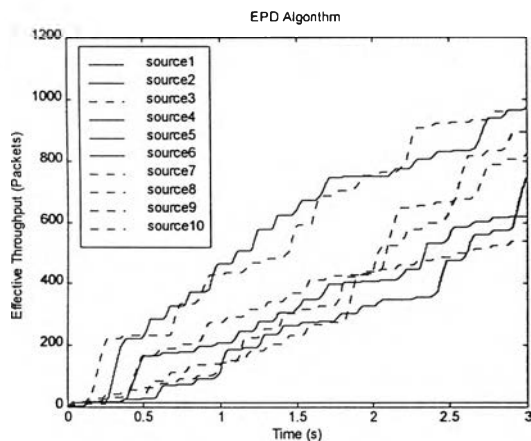
5.2.2 ผลการจำลองการทำงานโดยที่แหล่งกำเนิดเป็นแบบส่งต่อเนื่องต่อเนื่อง (persistent) และแบบเบิสต์ (burst)

การจำลองการทำงานในส่วนนี้จะทำการทดสอบกับแบบจำลอง Parking Lot โดยมีแหล่งกำเนิดที่เป็นแบบส่งต่อเนื่องและแบบส่งเป็นช่วงๆ ซึ่งในส่วนนี้จะเป็นการทดสอบถึงการปรับตัวในการส่งข้อมูลของแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่องเมื่อมีแหล่งกำเนิดที่ส่งข้อมูลเป็นแบบเบิสต์ แบบจำลองที่ใช้ในการทดสอบนี้เป็นแบบจำลองในรูปที่ 4.2 โดยมีข้อกำหนดของแบบจำลองดังในหัวข้อที่ 4.1.2 จากรูปที่ 4.2 แหล่งกำเนิดที่ 1 ถึง 3 เป็นแหล่งกำเนิดแบบเบิสต์ และแหล่งกำเนิดที่ 4 ถึง 10 เป็นแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่อง แหล่งกำเนิดที่ 7 ถึง 10 จะเป็นแหล่งกำเนิดที่ถูกแทรกเข้าไปในโครงข่าย โดยช่วงเวลาที่มีการส่งข้อมูลของแหล่งกำเนิดแบบเบิสต์จะส่งข้อมูล 100 KB และช่วงเวลาไม่ส่งข้อมูลของแหล่งกำเนิดแบบเบิสต์จะเป็นเวลา 1 วินาที ซึ่งแหล่งกำเนิดทั้งสองแบบนี้จะถูกควบคุมการส่งข้อมูลเข้าไปในโครงข่ายโดย TCP ซึ่งใช้ window เป็นกลไกในการควบคุมการส่งข้อมูล

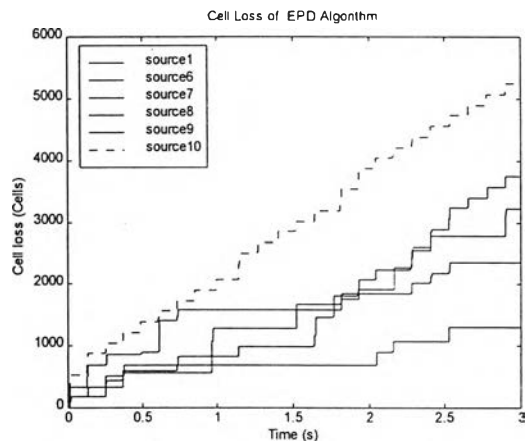
ตารางที่ 5.6 แสดงผลการจำลองการทำงานของอัลกอริทึม EPD ทั้ง 3 โดยมีแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่องและส่งข้อมูลเป็นแบบเบิสต์ ในโครงข่ายแบบ LAN รูปที่ 5.16 ถึง 5.18 แสดง Effective Throughput และ Cell loss ของอัลกอริทึมทั้ง 3 ที่มีแหล่งกำเนิดส่งข้อมูลต่อเนื่องและส่งข้อมูลแบบเบิสต์ ในโครงข่ายแบบ LAN

ตารางที่ 5.6 ผลการจำลองการทำงานของแบบจำลอง Parking Lot โดยมีแหล่งกำเนิดแบบส่งข้อมูลต่อเนื่อง และแบบเบิร์ตในโครงข่ายแบบ LAN

| Algorithm | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packet) | 5608 | 5432 | 6211 |
| TCP Throughput (Mbit/s) | 111.167 | 107.15 | 124.59 |
| Fairness index | 0.9629 | 0.9021 | 0.9923 |
| Efficiency (%) | 82.12 | 79.15 | 92.04 |

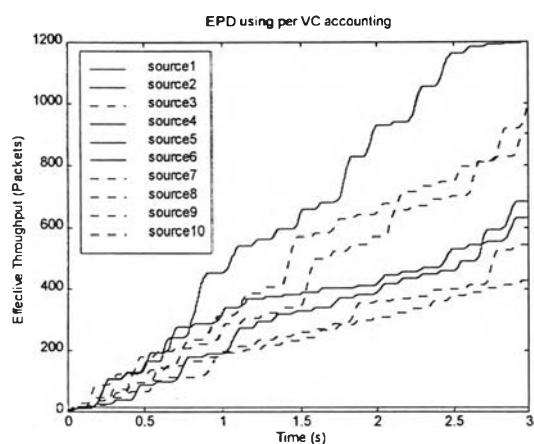


(ก)

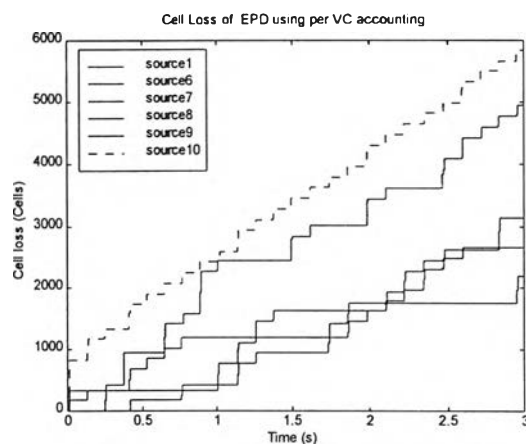


(ข)

รูปที่ 5.16 (ก) Effective Throughput (ข) Cell Loss ของอัลกอริทึม Early Packet Discard โดยที่แหล่งกำเนิด เป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิร์ต ในโครงข่าย LAN

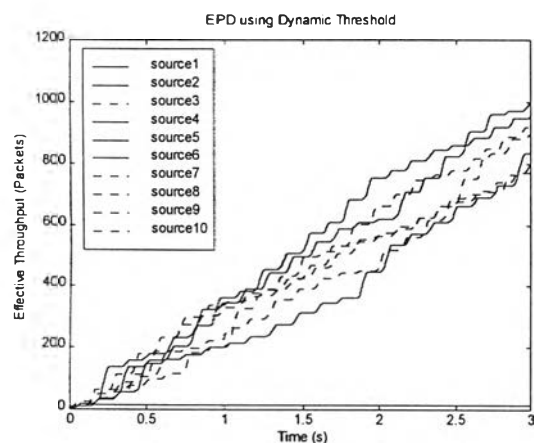


(ก)

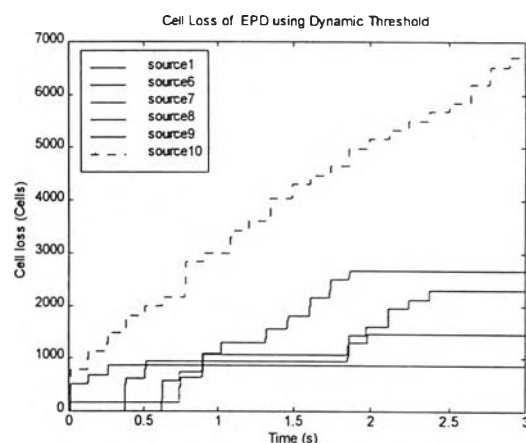


(ข)

รูปที่ 5.17 (ก) แสดง Effective Throughput (ข) Cell Loss ของอัลกอริทึม EPD ที่ใช้ per-VC accounting โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิรสต์ ในโครงข่าย LAN



(ก)



(ข)

รูปที่ 5.18 (ก) แสดง Effective Throughput (ข) Cell Loss ของอัลกอริทึม EPD ที่ใช้จุดเริ่มเปลี่ยนพลวัต โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิรสต์ ในโครงข่าย LAN

5.2.3 การวิเคราะห์ผลการจำลองแบบ Parking Lot

เมื่อแหล่งกำเนิดเป็นแบบส่งข้อมูลอย่างต่อเนื่องในโครงข่ายแบบ LAN จากผลการจำลอง อัลกอริทึม EPD จะพบว่า Effective Throughput ของแหล่งกำเนิดที่ 9 และ 10 มีค่าต่ำสุด ดังในรูปที่ 5.10 (ก) เพราะแหล่งกำเนิด 9 และ 10 จะถูกสวิตช์ทิ้งเซลล์ออกไปเป็นจำนวนมากที่สุด ดังในรูปที่ 5.10 (ข) สาเหตุที่ สวิตช์ทำการทิ้งเซลล์ของแหล่งกำเนิด 9 และ 10 ออกไปเป็นจำนวนมากเพราะแหล่งกำเนิดทั้ง 2 อยู่ใกล้ ปลายทางมากที่สุดทำให้ TCP สามารถตรวจจับแพ็กเก็ตที่สูญเสียได้มากกว่าแหล่งกำเนิดอื่นๆ แหล่งกำเนิด ทั้งสองจึงคอยแต่ทำการส่งแพ็กเก็ตที่สูญเสียใหม่อีกครั้งโดยไม่มีโอกาสส่งแพ็กเก็ตใหม่เพิ่มขึ้น

ผลการจำลองของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting จะมีค่า Effective Throughput ที่ มากขึ้นกว่าอัลกอริทึม EPD โดยเฉพาะ Effective Throughput ของแหล่งกำเนิดที่ 9 และ 10 จะมีค่ามากขึ้น ประมาณ 120 และ 60 แพ็กเก็ต ดังในรูปที่ 5.11 (ก) เพราะอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting มีการนับจำนวนเซลล์ที่อยู่ในบัฟเฟอร์ของสวิตช์และทำการทิ้งเซลล์ของแหล่งกำเนิด 9 น้อยลง ส่วนแหล่ง กำเนิดที่ 10 ยังคงมีการทิ้งเซลล์เป็นจำนวนมากเหมือนเดิม เพราะอยู่ใกล้ปลายทางมากกว่าแหล่งกำเนิดอื่น ทำให้มีโอกาสส่งข้อมูลได้เยอะกว่าจึงทำให้สวิตช์ทิ้งเซลล์ของแหล่งกำเนิด 10 ออกไปมากด้วยเช่นกัน ดังใน รูปที่ 5.11 (ข)

ผลการจำลองของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต จะมีค่า Effective Throughput และวิสัยสามารถมากที่สุด ดังในตารางที่ 5.4 จึงทำให้ประสิทธิภาพ TCP มากกว่าอัลกอริทึม EPD และ อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ประมาณ 7% และ 2% ตามลำดับ โดยเฉพาะแหล่งกำเนิดที่ 9 และ 10 มีจำนวนแพ็กเก็ตที่รับได้เพิ่มขึ้นเป็นจำนวน 300 แพ็กเก็ต เมื่อเทียบกับอัลกอริทึม EPD แต่จากรูป ที่ 5.12 (ก) แหล่งกำเนิดที่ 10 ก็ยังคงมี Effective Throughput น้อยกว่าแหล่งกำเนิดอื่นๆที่แทรกเข้ามา ระหว่างทาง เพราะแหล่งกำเนิดที่ 10 อยู่ใกล้ปลายทางมากที่สุดทำให้ปลายทางสามารถตรวจจับแพ็กเก็ตที่ สูญเสียได้เร็วกว่าแหล่งกำเนิดอื่นๆ ดังในรูปที่ 5.12 (ข) ซึ่งแหล่งกำเนิดที่ 10 จะมีจำนวนเซลล์ที่สูญเสียมาก กว่าแหล่งกำเนิดอื่น ดังนั้นจึงทำให้ค่า Effective Throughput ต่ำลง แต่อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่ม เปลี่ยนพลวัตจะมีการนับจำนวนเซลล์ที่อยู่ในสวิตช์และทำการทิ้งเซลล์ของแหล่งกำเนิดที่มีการใช้บัฟเฟอร์ มากกว่าระดับ fair share ออกไป จึงทำให้ค่าความเท่าเทียมกันของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยน พลวัตมีค่าเป็น 0.9951 ซึ่งเข้าใกล้ 1 มากที่สุด รองลงมาคืออัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting

สำหรับผลการจำลองการทำงานของแบบจำลอง Parking Lot ที่มีแหล่งกำเนิดเป็นแบบส่งต่อเนื่อง ต่อเนื่องและแบบเบิรสต์ พบว่าประสิทธิภาพ TCP ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีค่า มากที่สุด เพราะอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีการแบ่งแบนด์วิดท์ของแหล่งกำเนิดแบบ เบิรสต์ขณะไม่มีข้อมูลส่งไปให้แหล่งกำเนิดที่มีการใช้แบนด์วิดท์ต่ำ(ในที่นี้ก็คือแหล่งกำเนิดที่ 9 และ 10) ทำ

ให้แหล่งกำเนิดที่ 9 และ 10 สามารถส่งข้อมูลได้มากขึ้น ดังแสดงในรูปที่ 5.18 (ก) จึงเป็นสาเหตุให้ประสิทธิภาพและความเท่าเทียมกันดีขึ้นกว่าอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ซึ่งอัลกอริทึมทั้งสองเมื่อมีแหล่งกำเนิดแบบเบิรสต์อยู่ในโครงข่ายจะทำให้ประสิทธิภาพและความเท่าเทียมกันในการส่งข้อมูลลดลง เพราะช่วงเวลาส่งข้อมูลและหยุดส่งข้อมูลของแหล่งกำเนิดแบบเบิรสต์ทำให้แหล่งกำเนิดแบบส่งต่อเนื่องต่อเนื่องปรับตัวในการส่งข้อมูลไม่ทันเมื่อแหล่งกำเนิดแบบเบิรสต์มีการเปลี่ยนช่วงเวลาจากช่วงส่งข้อมูลเป็นช่วงเวลาหยุดส่งข้อมูลหรือจากช่วงเวลาหยุดส่งข้อมูลไปเป็นช่วงเวลาส่งข้อมูล

สรุปจากผลการจำลองของแบบจำลอง Parking Lot ในโครงข่ายแบบ LAN อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตและอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting สามารถปรับปรุงสมรรถนะของ TCP ได้ทั้งในกรณีของวิสัยสามารถและความเท่าเทียม ส่วนผลการจำลองของแบบจำลอง Parking Lot ที่มีแหล่งกำเนิดเป็นแบบส่งต่อเนื่องต่อเนื่องและแบบเบิรสต์ อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตสามารถปรับปรุงสมรรถนะของ TCP ได้ทั้งวิสัยสามารถและความเท่าเทียมกัน โดยเฉพาะความเท่าเทียมกันมีค่าเข้าใกล้ 1 มากแสดงว่าแต่ละแหล่งกำเนิดมีความเท่าเทียมกันในการส่งข้อมูล

5.3 ผลการจำลองการทำงานของแบบจำลอง Chain

5.3.1 ผลการจำลองการทำงานโดยที่แหล่งกำเนิดเป็นแบบส่งต่อเนื่อง (persistent) ทั้งหมด ในโครงข่าย

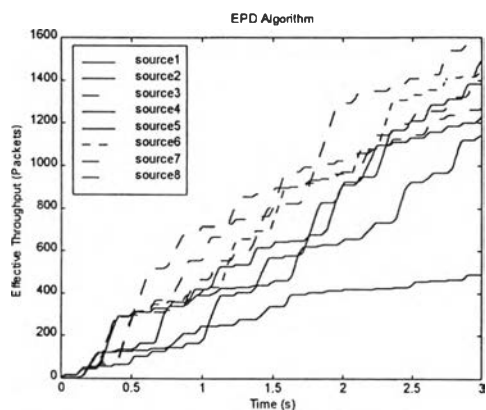
Local Area Network (LAN) และ Wide Area Network (WAN)

ในส่วนนี้จะทำการจำลองการทำงานของแบบจำลอง Chain ดังแสดงในรูปที่ 4.3 โดยแบบจำลอง Chain จะมีแหล่งกำเนิดที่ผ่านหลาย link ซึ่งมีประวิงเวลามากกว่าแหล่งกำเนิดที่ผ่าน link เดียว ซึ่งแบบจำลอง Chain นี้เป็นแบบจำลองหนึ่ง queuing ถึงผลกระทบของแหล่งกำเนิดที่มีค่าประวิงเวลาไม่เท่ากัน นอกจากนี้แบบจำลอง Chain ยังเป็นแบบจำลองที่ทำการศึกษาการแทรกแหล่งกำเนิดเข้ามาและการออกของแหล่งกำเนิดในระหว่างทาง ในการจำลองใช้แหล่งกำเนิดที่มีการส่งข้อมูลอย่างต่อเนื่อง โดย TCP จะทำการควบคุมการส่งข้อมูลเข้าไปในโครงข่ายของแหล่งกำเนิดอีกครั้งหนึ่ง ในการจำลองนี้ให้ขนาดบัฟเฟอร์มีค่า 2000 เซลล์ ระดับจุดเริ่มเปลี่ยนของอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting มีค่าเป็น 1600 เซลล์ (60%ของขนาดบัฟเฟอร์) และระดับจุดเริ่มเปลี่ยนด้านสูง (High Threshold) และด้านต่ำ (Low Threshold) ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนแบบพลวัตมีค่าเป็น 1600 และ 1000 เซลล์ ตามลำดับ ส่วนข้อกำหนดอื่นๆของแบบจำลอง Chain เป็นไปดังในส่วนที่ 4.1.3

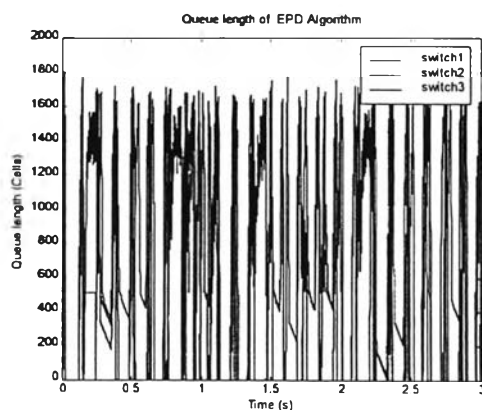
ในตารางที่ 5.7 แสดงผลการจำลองของอัลกอริทึมทั้ง 3 โดยที่แหล่งกำเนิดมีการส่งข้อมูลต่อเนื่องในโครงข่ายแบบ LAN ตารางที่ 5.8 แสดงผลการทำงานของอัลกอริทึมทั้ง 3 โดยที่แหล่งกำเนิดมีการส่งข้อมูลต่อเนื่อง ในโครงข่ายแบบ WAN รูปที่ 5.19 ถึง 5.21 แสดง Effective Throughput ของอัลกอริทึมทั้ง 3 ที่มีแหล่งกำเนิดส่งข้อมูลต่อเนื่องในโครงข่ายแบบ LAN รูปที่ 5.22 ถึง 5.24 แสดง Effective Throughput ของอัลกอริทึมทั้ง 3 ที่มีแหล่งกำเนิดส่งข้อมูลต่อเนื่องในโครงข่ายแบบ WAN

ตารางที่ 5.7 ผลการจำลองการทำงานของแบบจำลอง Chain โดยที่มีแหล่งกำเนิดส่งข้อมูลอย่างต่อเนื่อง ในโครงข่ายแบบ LAN

| Algorithm | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packet) | 10049 | 9401 | 9798 |
| TCP Throughput (Mbit/s) | 91.692 | 94.454 | 97.582 |
| Fairness index | 0.9385 | 0.9698 | 0.9914 |
| Efficiency (%) | 67.73 | 69.77 | 72.09 |

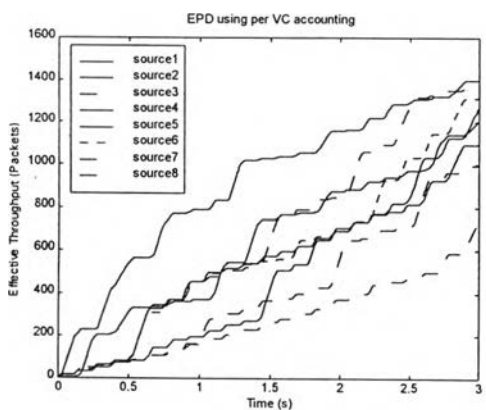


(ก)

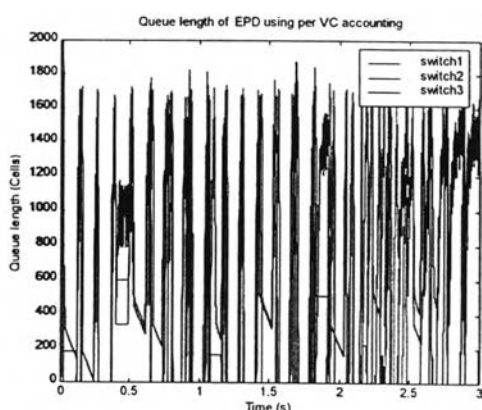


(ข)

รูปที่ 5.19 (ก) Effective Throughput (ข) ความยาวคิว ของอัลกอริทึม Early Packet Discard โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่าย LAN

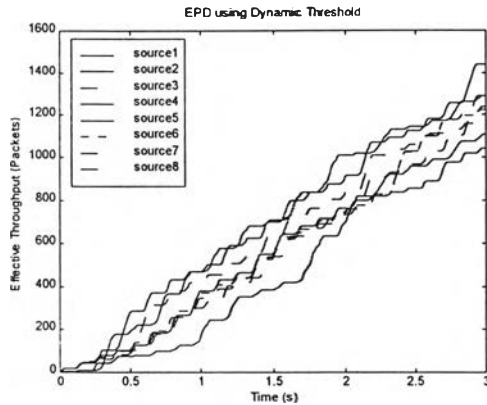


(ก)

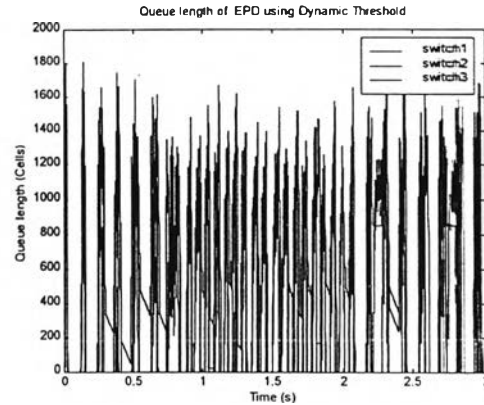


(ข)

รูปที่ 5.20 (ก) แสดง Effective Throughput (ข) ความยาวคิว ของอัลกอริทึม EPD ที่ใช้ per-VC accounting โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่าย LAN



(ก)

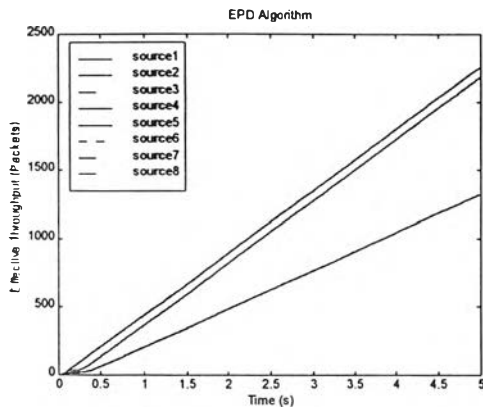


(ข)

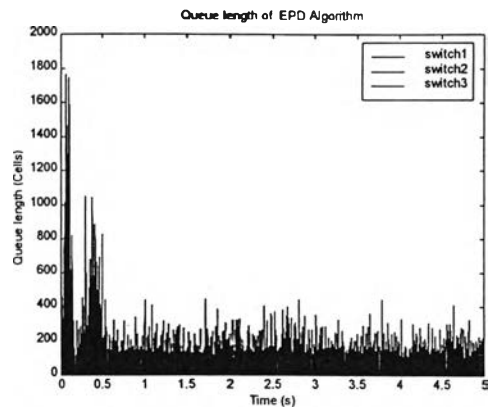
รูปที่ 5.21 (ก) Effective Throughput (ข) ความยาวคิว ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องในโครงข่าย LAN

ตารางที่ 5.8 ผลการจำลองการทำงานของแบบจำลอง Chain โดยที่มีแหล่งกำเนิดส่งข้อมูลอย่างต่อเนื่อง ในโครงข่ายแบบ WAN

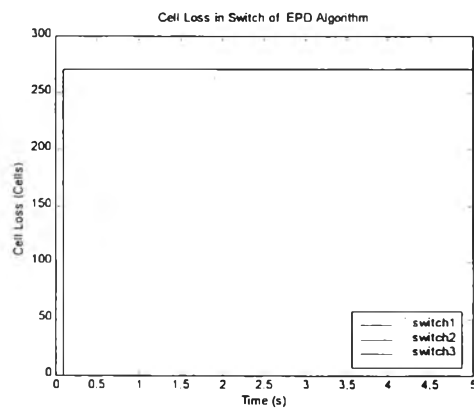
| Algorithm | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packet) | 16068 | 16227 | 16154 |
| TCP Throughput (Mbit/s) | 88.493 | 90.359 | 89.504 |
| Fairness index | 0.9620 | 0.9614 | 0.9620 |
| Efficiency (%) | 65.37 | 66.75 | 66.12 |



(ก)

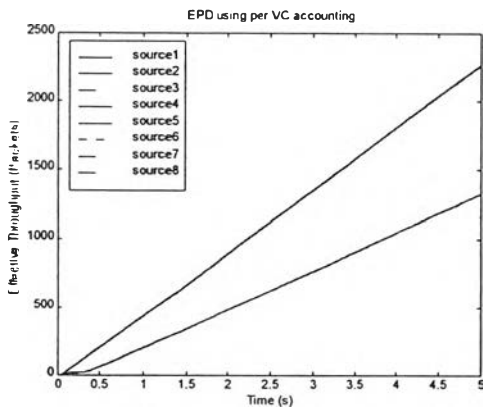


(ข)

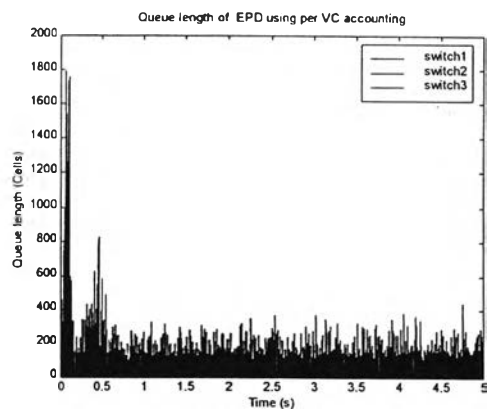


(ค)

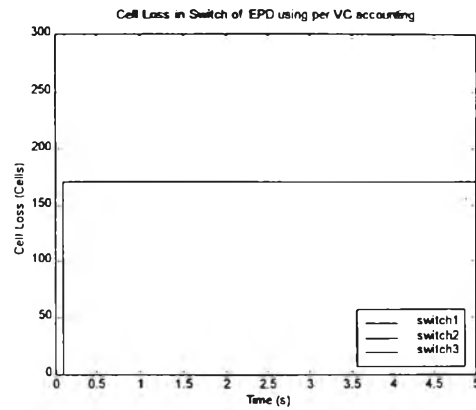
รูปที่ 5.22 (ก) Effective Throughput (ข) ความยาวคิว (ค) Cell Loss ของอัลกอริทึม Early Packet Discard โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่าย WAN



(ก)

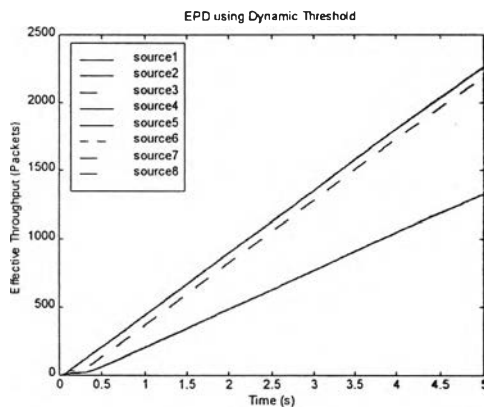


(ข)

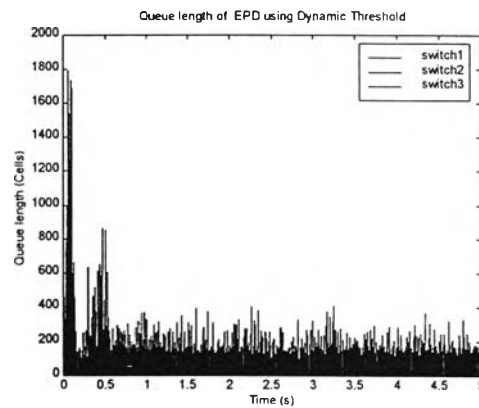


(ค)

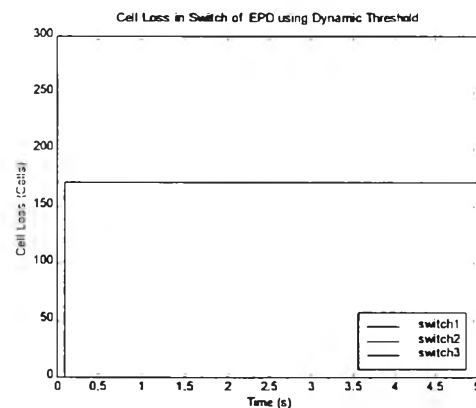
รูปที่ 5.23 (ก) แสดง Effective Throughput (ข) ความยาวคิว (ค) Cell Loss ของอัลกอริทึม EPD ที่ใช้ per-VC accounting โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่อง ในโครงข่าย WAN



(ก)



(ข)



(ค)

รูปที่ 5.24 (ก) Effective Throughput (ข) ความยาวคิว (ค) Cell Loss ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่ม เปลี่ยนพลวัตโดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องในโครงข่าย WAN

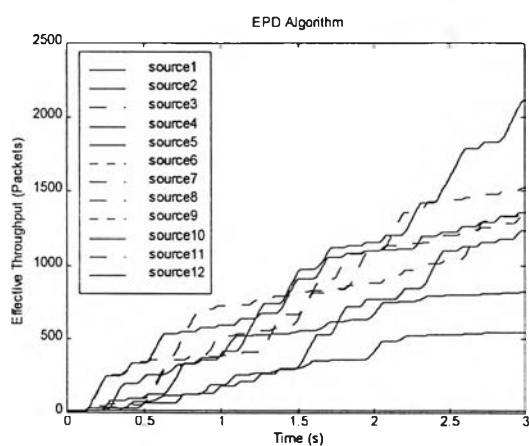
5.3.2 ผลการจำลองการทำงานของแบบจำลอง Chain โดยมีแหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิรสต์

ในส่วนนี้ทำการทดสอบการทำงานของอัลกอริทึมทั้ง 3 กับแบบจำลอง Chain ที่มีแหล่งกำเนิดเป็นแบบส่งข้อมูลอย่างต่อเนื่องและแบบเบิรสต์ แบบจำลอง Chain ที่ใช้แสดงในรูปที่ 4.4 โดยมีแหล่งกำเนิดทั้งหมด 12 แหล่งกำเนิด ซึ่งแหล่งกำเนิดที่ 1,2,4,5,7,8,10 และ 11 เป็นแบบส่งข้อมูลอย่างต่อเนื่องตลอดเวลา ส่วนแหล่งกำเนิดที่ 3, 6, 9 และ 12 เป็นแบบเบิรสต์ แหล่งกำเนิดที่เป็นแบบเบิรสต์จะมีช่วงเวลาที่ส่งข้อมูลและหยุดส่งข้อมูล โดยช่วงเวลาที่ส่งข้อมูลได้จะส่งข้อมูลเป็นจำนวน 100 KB และทำการหยุดส่งเป็นเวลา 1 วินาที จากนั้นจึงกลับมาส่งข้อมูลใหม่ แบบจำลอง Chain ใช้ทดสอบในโครงข่ายแบบ LAN

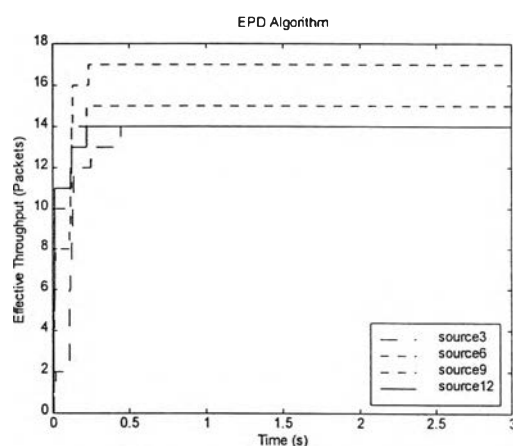
ตารางที่ 5.9 แสดงผลการจำลองการทำงานของอัลกอริทึม EPD ทั้ง 3 บนแบบจำลอง Chain โดยมีแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่องตลอดเวลาและส่งข้อมูลเป็นแบบเบิรสต์ในโครงข่ายแบบ LAN รูปที่ 5.25 ถึง 5.27 แสดง Effective Throughput และ Cell loss ของอัลกอริทึมทั้ง 3 ที่มีแหล่งกำเนิดส่งข้อมูลต่อเนื่องและส่งข้อมูลแบบเบิรสต์ในโครงข่ายแบบ LAN

ตารางที่ 5.9 ผลการจำลองการทำงานของแบบจำลอง Chain โดยมีแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่องและแบบเบิร์ต

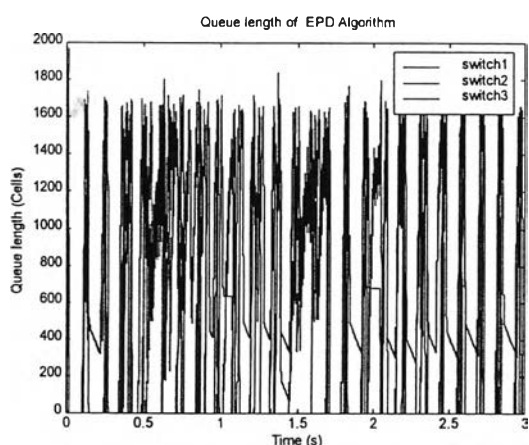
| Algorithm | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packet) | 10347 | 9847 | 10753 |
| TCP Throughput (Mbit/s) | 77.357 | 82.76 | 113.31 |
| Fairness index | 0.8957 | 0.9246 | 0.9847 |
| Efficiency (%) | 57.15 | 61.14 | 83.71 |



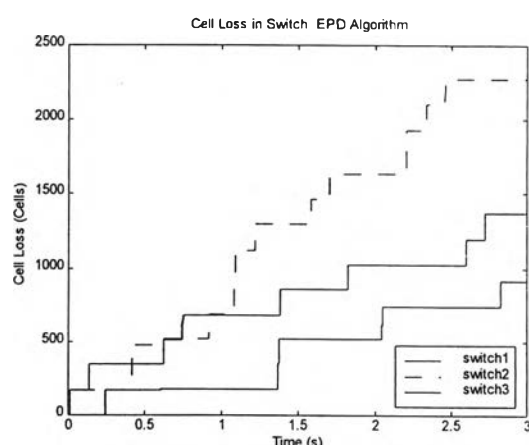
(ก)



(ข)

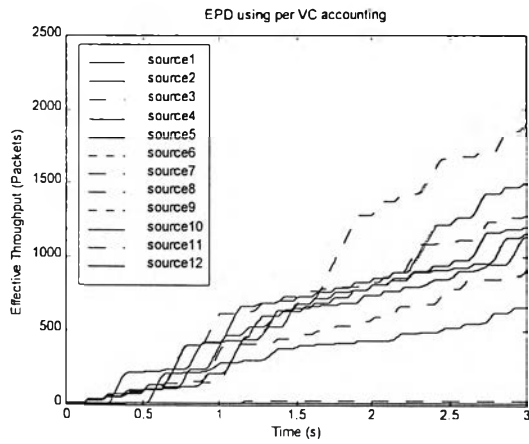


(ค)

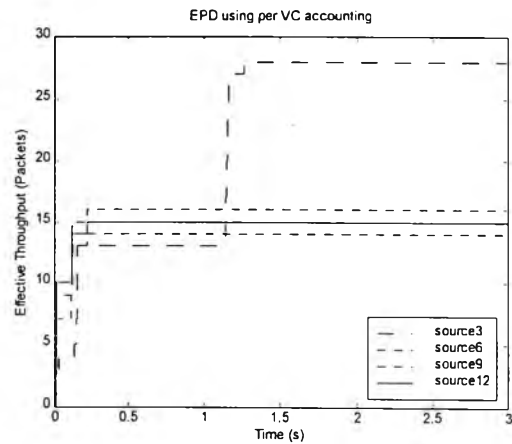


(ง)

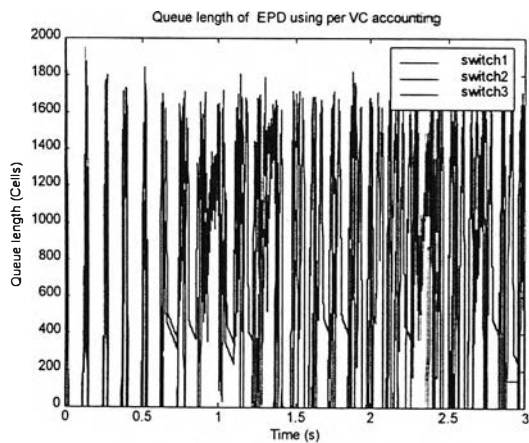
รูปที่ 5.25 (ก),(ข) Effective Throughput (ค) ความยาวคิว (ง) Cell Loss ของอัลกอริทึม Early Packet Discard โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิร์ต ในโครงข่ายแบบ LAN



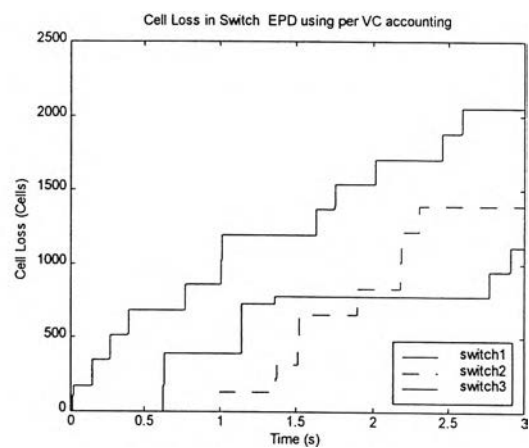
(ก)



(ข)

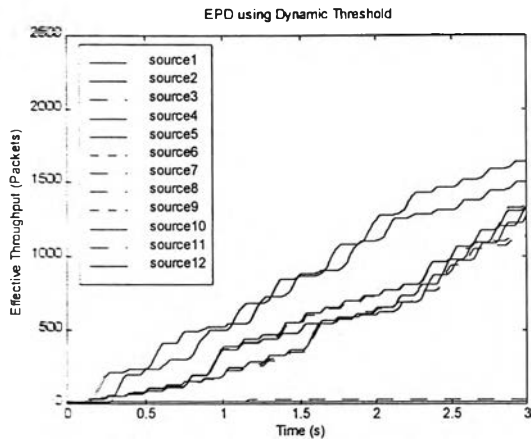


(ค)

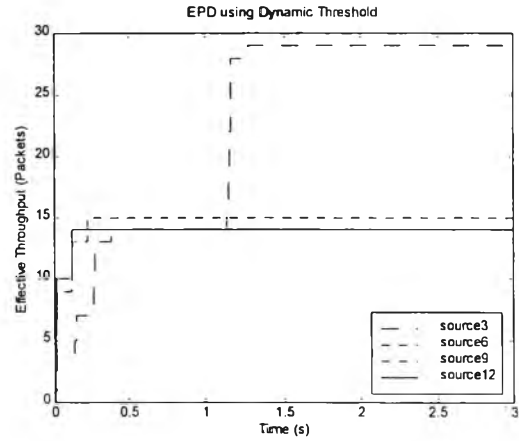


(ง)

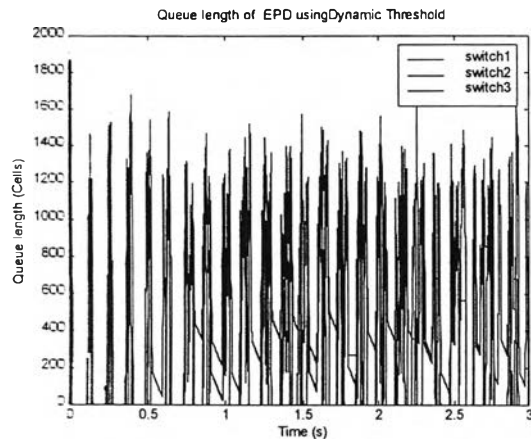
รูปที่ 5.26 (ก),(ข) Effective Throughput (ค) ความยาวคิว (ง) Cell Loss ของอัลกอริทึม EPD ที่ใช้ per-VC accounting โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิรสต์ ในโครงข่ายแบบ LAN



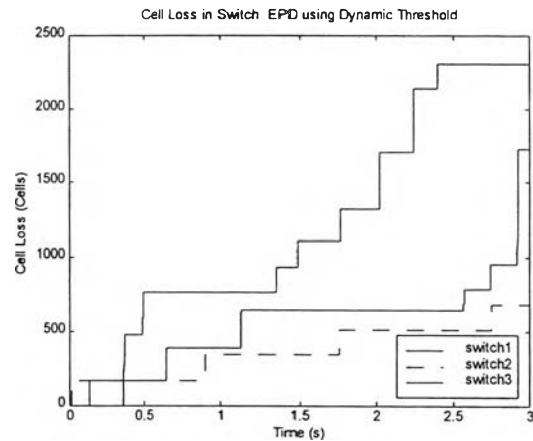
(ก)



(ข)



(ค)



(ง)

รูปที่ 5.27 (ก),(ข) Effective Throughput (ค) ความยาวคิว (ง) Cell Loss ของอัลกอริทึม EPD ที่ใช้จุดเริ่มเปลี่ยนพลวัต โดยที่แหล่งกำเนิดเป็นแบบส่งข้อมูลต่อเนื่องและแบบเบิรสต์ ในโครงข่ายแบบ LAN

5.3.3 การวิเคราะห์ผลการจำลองของแบบจำลอง Chain

เมื่อโครงข่ายเป็นแบบ LAN จากผลการจำลองอัลกอริทึม EPD พบว่าแหล่งกำเนิดที่ 1 และ 2 ดังในรูปที่ 5.19 (ก) ซึ่งมีค่าประวิงเวลามากที่สุดเพราะต้องส่งข้อมูลผ่านหลาย hop มีค่า Effective Throughput น้อยกว่าแหล่งกำเนิดอื่นๆ ซึ่งมีค่าประวิงเวลาน้อยและส่งข้อมูลผ่านเพียง 1 hop เท่านั้น ด้วยสาเหตุนี้จึงเป็นผลทำให้อัลกอริทึม EPD มีค่าความเท่าเทียมกัน (fairness) ในการส่งข้อมูลต่ำกว่าอัลกอริทึมอื่นๆ ไปด้วย ส่วนอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting จะมีการตรวจจับข้อมูลที่ส่งเข้ามาที่สวิตช์ของแต่ละแหล่งกำเนิด ถ้าแหล่งกำเนิดใดมีจำนวนเซลล์มากกว่าค่า fair share สวิตช์ก็จะทำการทิ้งเซลล์ของแหล่งกำเนิดนั้นออกไป จึงทำให้มีค่าความเท่าเทียมกันมากกว่าอัลกอริทึม EPD ด้วยเหตุนี้จึงทำให้แหล่งกำเนิดที่ 1 และ 2 มีค่า Effective Throughput มากขึ้นกว่าอัลกอริทึม EPD

ในกรณีอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต จะมีค่าความเท่าเทียมกันและประสิทธิภาพ TCP สูงกว่าอัลกอริทึมทั้ง 2 จากรูปที่ 5.21 (ก) แหล่งกำเนิดที่ 1 และ 2 จะมีค่า Effective Throughput ไม่น้อยกว่าแหล่งกำเนิดอื่นๆ เนื่องจากอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตจะมีการตรวจจับข้อมูลที่เข้ามาในสวิตช์ของแต่ละแหล่งกำเนิดเหมือนกับอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting เมื่อพิจารณาที่สวิตช์ 1 ซึ่งมีแหล่งกำเนิดที่ 1 ถึง 4 เชื่อมต่ออยู่ สวิตช์จะทำการทิ้งเซลล์ออกไปเมื่อแหล่งกำเนิดส่งข้อมูลเข้าไปในสวิตช์มากเกินไปกว่าระดับ fair share จึงทำให้แหล่งกำเนิดที่ 1 ถึง 4 ส่งข้อมูลได้ใกล้เคียงกันเมื่อเทียบกับอัลกอริทึม EPD และจุดเริ่มเปลี่ยนพลวัตจะช่วยทำให้แหล่งกำเนิดที่ 1 และ 2 ซึ่งมีค่าประวิงเวลามากสามารถส่งข้อมูลได้มากขึ้น โดยนำเอาแบนด์วิดท์ที่เหลือจากแหล่งกำเนิดที่ 3 และ 4 มาแบ่งให้กับแหล่งกำเนิดที่ 1 และ 2 (กรณีนี้จะเกิดขึ้นเมื่อแหล่งกำเนิดที่ 3 และ 4 ถูกทิ้งเซลล์ออกไปจากสวิตช์ทำให้ขณะนั้นมีแบนด์วิดท์เหลือ)

ในโครงข่ายแบบ WAN จากผลการจำลอง ไม่ว่าอัลกอริทึมที่สวิตช์จะเป็นแบบใดประสิทธิภาพ TCP ในโครงข่ายก็ยังไม่มีความแตกต่างกันมากนัก เนื่องจากในโครงข่ายแบบ WAN จะมีระยะทางไกลจึงทำให้แต่ละแหล่งกำเนิดจะสามารถส่งข้อมูลใหม่ได้หรือส่งแพ็กเก็ตที่ถูกทิ้งออกไปที่สวิตช์ใหม่อีก เมื่อได้รับ ACK จากปลายทางก่อนถึงจะส่งข้อมูลได้ ซึ่งกว่าที่แหล่งกำเนิดจะได้รับ ACK โครงข่ายก็ไม่ได้ความคับคั่งแล้วจึงทำให้โครงข่ายแบบ WAN มีค่า Effective Throughput มากกว่าในโครงข่ายแบบ LAN

สำหรับอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตสามารถปรับปรุงประสิทธิภาพเพียงเล็กน้อยส่วนค่าความเท่าเทียมกันจะเท่ากับอัลกอริทึม EPD ที่เป็นเช่นนี้เพราะว่าอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตจะมีการทิ้งแพ็กเก็ตออกไปเป็นจำนวนมากเพื่อปรับปรุงความเท่าเทียมกันแต่ในโครงข่ายแบบ WAN มีค่าประวิงเวลามากจึงทำให้ปลายทางตรวจจับแพ็กเก็ตที่สูญเสียนั้นที่ถูกทิ้งออกไปได้ช้าและเมื่อส่ง ACK กลับมาออกต้นทาง โดยทางต้นทางก็จะได้รับ ACK ช้าด้วยเพราะระยะทางไกลกว่า ACK จากปลายทาง

ทางมาถึงต้นทาง เป็นผลทำให้แหล่งกำเนิดทำการส่งข้อมูลใหม่ได้ช้าด้วย ขณะที่บางแหล่งกำเนิดถูกทิ้งเซลล์ออกไปที่สวิตช์แต่บางแหล่งกำเนิดไม่ถูกทิ้งเซลล์เมื่อแหล่งกำเนิดที่ถูกทิ้งเซลล์มีการส่งแพ็กเก็ตที่ถูกทิ้งออกไปใหม่ช้าทำให้เสียเปรียบแหล่งกำเนิดที่ไม่ถูกทิ้งเซลล์ออกไป ซึ่งแหล่งกำเนิดที่ไม่ถูกทิ้งเซลล์ก็สามารถส่งข้อมูลได้ต่อไปเรื่อยๆ ด้วยเหตุนี้จึงเป็นสาเหตุให้อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตไม่สามารถปรับปรุงวิสัยความสามารถและความเท่าเทียมกันในการส่งข้อมูล

ในส่วนผลการจำลองของแบบจำลอง Chain โดยที่มีแหล่งกำเนิดแบบส่งข้อมูลต่อเนื่องตลอดเวลา และแบบเบิรสต์ พบว่าประสิทธิภาพและความเท่าเทียมกันของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมากกว่าอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting เพราะอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีการแบ่งแบนด์วิดท์ของแหล่งกำเนิดแบบเบิรสต์ในขณะที่ไม่ส่งข้อมูลให้กับแหล่งกำเนิดที่ส่งข้อมูลต่อเนื่องตลอดเวลา และมีการตรวจนับจำนวนเซลล์ที่อยู่ในบัฟเฟอร์ไม่ให้เกิดระดับ fair share ซึ่งเหตุผลทั้งสองนี้เป็นเหตุผลหลักที่ทำให้อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตสามารถปรับปรุงสมรรถนะ TCP ของอัลกอริทึม EPD ได้ดี ส่วนอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting เป็นอัลกอริทึมที่มีการนับจำนวนเซลล์ที่อยู่ในบัฟเฟอร์จึงทำให้ความเท่าเทียมกันมากกว่าอัลกอริทึม EPD จากรูปที่ 5.25 (ข), 5.26 (ข) และ 5.27 (ข) มีบางแหล่งกำเนิดแบบเบิรสต์ที่สามารถส่งข้อมูลได้ครั้งเดียว แต่บางแหล่งกำเนิดแบบเบิรสต์สามารถส่งข้อมูลได้ 2 ครั้ง

5.4 ผลการปรับระดับจุดเริ่มเปลี่ยนด้านสูง (High Threshold) และด้านต่ำ (Low Threshold)

ผลการจำลองในส่วนนี้เป็นการศึกษาถึงผลกระทบในการปรับระดับจุดเริ่มเปลี่ยนของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต โดยเปรียบเทียบกับอัลกอริทึม EPD และ อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting บนแบบจำลอง Peer to Peer ในโครงข่ายแบบ LAN โดยจุดเริ่มเปลี่ยนของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีหลายจุดเริ่มเปลี่ยน เช่น จุดเริ่มเปลี่ยนด้านสูง (High Threshold) จุดเริ่มเปลี่ยนด้านต่ำ (Low Threshold) ระดับ fair share และจุดเริ่มเปลี่ยนพลวัต แต่ระดับ fair share และจุดเริ่มเปลี่ยนพลวัต เป็นจุดเริ่มเปลี่ยนที่สวิตช์จะเป็นตัวคำนวณหาอัตรา โนมัตเมื่อมีแพ็กเกจใหม่เข้ามาที่สวิตช์ ดังนั้นจุดเริ่มเปลี่ยนที่สามารถปรับได้คือ จุดเริ่มเปลี่ยนด้านสูงและจุดเริ่มเปลี่ยนด้านต่ำ โดยในการจำลองนี้จะทำการปรับทีละจุดเริ่มเปลี่ยนอันใดอันหนึ่ง และทำการตั้งค่าอีกจุดเริ่มเปลี่ยนหนึ่งให้คงที่

ในส่วนแรกจะทำการปรับจุดเริ่มเปลี่ยนด้านสูงและกำหนดจุดเริ่มเปลี่ยนด้านต่ำให้มีค่าเป็น 1000 เซลล์ (50% ของขนาดบัฟเฟอร์) แล้วเปรียบเทียบค่าประสิทธิภาพ TCP และค่าความเท่าเทียมกันกับอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ในส่วนที่สองจะเป็นการปรับจุดเริ่มเปลี่ยนด้านต่ำและกำหนดจุดเริ่มเปลี่ยนด้านสูงไว้ที่ค่า 1900 เซลล์ (95% ของขนาดบัฟเฟอร์) แล้วทำการเปรียบเทียบกับอัลกอริทึมทั้ง 2 โดยตารางที่ 5.10 แสดงประสิทธิภาพ TCP เมื่อมีการปรับจุดเริ่มเปลี่ยนด้านสูงและกำหนดจุดเริ่มเปลี่ยนด้านต่ำไว้ที่ 1000 เซลล์ ตารางที่ 5.11 แสดงค่าความเท่าเทียมกันในการส่งข้อมูล เมื่อมีการปรับจุดเริ่มเปลี่ยนด้านสูงและกำหนดจุดเริ่มเปลี่ยนด้านต่ำไว้ที่ 1000 เซลล์ ในตารางที่ 5.12 แสดงค่าความเท่าเทียมกันและประสิทธิภาพ TCP เมื่อทำการปรับจุดเริ่มเปลี่ยนด้านต่ำและกำหนดจุดเริ่มเปลี่ยนด้านสูงไว้ที่ขนาด 1900 เซลล์

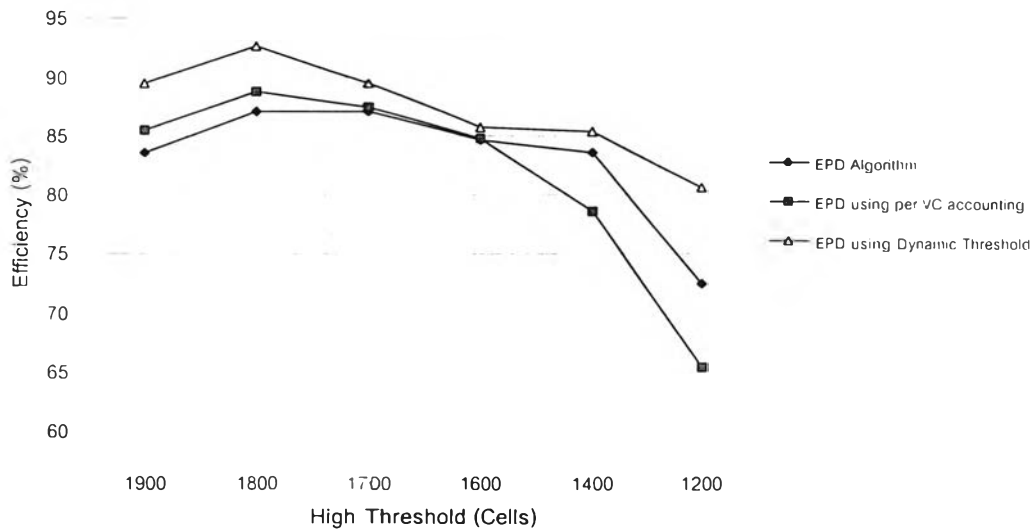
◆ ผลการปรับจุดเริ่มเปลี่ยนด้านสูงของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

ตารางที่ 5.10 ประสิทธิภาพ TCP โดยการปรับจุดเริ่มเปลี่ยนด้านสูงและให้จุดเริ่มเปลี่ยนด้านต่ำคงที่

| Threshold | Early Packet Discard (ERD) | ERD using per VC accounting | ERD using Dynamic Threshold |
|-----------|----------------------------|-----------------------------|-----------------------------|
| 95% | 83.59 | 85.58 | 89.54 |
| 90% | 87.14 | 88.71 | 92.63 |
| 85% | 87.09 | 87.45 | 89.54 |
| 80% | 84.66 | 84.8 | 85.78 |
| 70% | 83.63 | 78.52 | 85.47 |
| 60% | 72.49 | 65.55 | 80.63 |

ตารางที่ 5.11 ค่าความเท่าเทียมกัน โดยการปรับจุดเริ่มเปลี่ยนด้านสูงและให้จุดเริ่มเปลี่ยนด้านต่ำคงที่

| Threshold | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
|-----------|----------------------------|-----------------------------|-----------------------------|
| 95% | 0.9436 | 0.9526 | 0.9637 |
| 90% | 0.9326 | 0.9655 | 0.9673 |
| 85% | 0.9354 | 0.9786 | 0.9869 |
| 80% | 0.8682 | 0.9606 | 0.9615 |
| 70% | 0.9576 | 0.9588 | 0.9844 |
| 60% | 0.9797 | 0.9534 | 0.9820 |



รูปที่ 5.28 ผลการเปรียบเทียบประสิทธิภาพ TCP เมื่อปรับจุดเริ่มเปลี่ยนด้านสูง

◆ ผลการปรับจุดเริ่มเปลี่ยนด้านต่ำของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

ตารางที่ 5.12 ประสิทธิภาพของ TCP โดยกำหนดจุดเริ่มเปลี่ยนด้านสูงให้คงที่และปรับจุดเริ่มเปลี่ยนด้านต่ำ

| Low Threshold (cells) | Throughput (Mbit/s) | Fairness | Efficiency (%) |
|-----------------------------|---------------------|----------|----------------|
| 1800 (90%) | 119.65 | 0.9428 | 88.39 |
| 1700 (85%) | 123.79 | 0.9526 | 91.45 |
| 1600 (80%) | 125.12 | 0.9586 | 92.43 |
| 1400 (70%) | 122.16 | 0.9609 | 90.24 |
| 1200 (60%) | 121.81 | 0.9936 | 89.99 |
| 1000 (50%) | 121.11 | 0.9949 | 89.46 |
| EPD Algorithm | 113.16 | 0.9436 | 83.59 |
| EPD using per VC accounting | 115.852 | 0.9526 | 85.58 |

การวิเคราะห์ผลการปรับระดับจุดเริ่มเปลี่ยน

จากผลการจำลองการปรับระดับจุดเริ่มเปลี่ยน EPD และจุดเริ่มเปลี่ยนด้านสูง (อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต เมื่อนำระดับจุดเริ่มเปลี่ยน EPD และจุดเริ่มเปลี่ยนด้านสูงมาเปรียบเทียบกับ จะพบว่าประสิทธิภาพ TCP ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตสูงกว่าประสิทธิภาพของอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ไม่ว่าจะระดับของจุดเริ่มเปลี่ยนจะเป็นเท่าใด เพราะอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีการนำเอาแบนด์วิดท์ที่เหลือจาก VC ที่มีการใช้งานแบนด์วิดท์ต่ำมาแบ่งให้กับ VC ที่มีความต้องการใช้แบนด์วิดท์สูง ซึ่งดีกว่าทั้งแบนด์วิดท์ที่ไม่ใช้งานไว้โดยเปล่าประโยชน์

จากตารางที่ 5.10 ที่ระดับจุดเริ่มเปลี่ยน 1900 เซลล์ (95% ของขนาดบัฟเฟอร์) ทั้ง 3 อัลกอริทึมจะมีประสิทธิภาพลดลงเนื่องจากที่ระดับจุดเริ่มเปลี่ยน 1900 เซลล์ มีขนาดช่องว่างระหว่างระดับจุดเริ่มเปลี่ยนกับขนาดของบัฟเฟอร์น้อยเกินไป ทำให้มีเซลล์ล้นออกไปจากบัฟเฟอร์ จึงทำให้มีแพ็กเก็ตที่เสียเกิดขึ้นในโครงข่าย ซึ่งแพ็กเก็ตที่เสียนี้ไปถึงปลายทางแล้ว ปลายทางไม่สามารถใช้งานแพ็กเก็ตนั้นได้ จึงต้องมีการส่งแพ็กเก็ตนั้นใหม่ทำให้สูญเสียแบนด์วิดท์ในโครงข่ายและทำให้ประสิทธิภาพลดลง ที่ระดับจุดเริ่มเปลี่ยน EPD และจุดเริ่มเปลี่ยนด้านสูงเป็น 1800 เซลล์ (90% ของขนาดบัฟเฟอร์) ประสิทธิภาพ TCP ของทั้ง 3 อัลกอริทึมมีค่ามากที่สุดเนื่องจากการใช้งานบัฟเฟอร์ได้อย่างเต็มประสิทธิภาพ และระดับจุดเริ่มเปลี่ยน 1800 เซลล์เป็นระดับที่มีเซลล์ล้นออกจากบัฟเฟอร์น้อยเมื่อเปรียบเทียบกับที่ระดับจุดเริ่มเปลี่ยน 1900 เซลล์ ส่วนระดับจุดเริ่มเปลี่ยน EPD และจุดเริ่มเปลี่ยนด้านสูงในระดับอื่นๆที่ต่ำกว่าระดับ 1800 เซลล์จะมีค่าประสิทธิภาพลดลงไปเรื่อยๆ เนื่องจากบัฟเฟอร์ของสวิตช์มีการใช้งานต่ำลงเรื่อยๆ

จากตารางที่ 5.11 แสดงค่าความเท่าเทียมกัน โดยจะพบว่าอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตจะมีค่าความเท่าเทียมกันมากกว่าอัลกอริทึมอื่นๆ เนื่องจากอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีการนับจำนวนเซลล์ที่อยู่ในบัฟเฟอร์และทิ้งเซลล์ออกไปเมื่อมีจำนวนเซลล์มากเกินกว่าระดับ fair share ด้วยเหตุนี้จึงทำให้อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีค่า fairness มากกว่าอัลกอริทึมอื่นๆ โดยอัลกอริทึม EPD จะไม่มีการนับเซลล์ที่อยู่ในบัฟเฟอร์ฉะนั้นอัลกอริทึม EPD จึงไม่รู้ว่า VC ใดมีการใช้งานบัฟเฟอร์มากกว่ากัน เมื่อเกิดความคับคั่งสวิตช์จึงไม่รู้ว่าควรทิ้งเซลล์ของ VC ใดออกไปก่อน จึงทำให้อัลกอริทึม EPD ทำการทิ้งเซลล์แบบสุ่ม จึงเป็นสาเหตุให้ค่าความเท่าเทียมกันต่ำกว่าอัลกอริทึมอื่นๆ

จากผลการจำลองการเปลี่ยนระดับจุดเริ่มเปลี่ยนด้านต่ำของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต ในตารางที่ 5.12 พบว่าเมื่อเปลี่ยนระดับจุดเริ่มเปลี่ยนด้านต่ำจะไม่มีผลต่อการเปลี่ยนแปลงค่าประสิทธิภาพมากนักแต่จะมีผลต่อค่าความเท่าเทียมกันในการส่งข้อมูลมากกว่า โดยความแตกต่างระหว่างระดับของจุดเริ่มเปลี่ยนด้านสูงกับจุดเริ่มเปลี่ยนด้านต่ำมีระดับที่แตกต่างกันมากจะทำให้ความเท่าเทียมกัน

ในการส่งข้อมูลของแต่ละแหล่งกำเนิดมีค่ามากจนเกือบจะเป็น 1 เพราะทำให้มีช่วงของเงื่อนไขที่สวิทช์จะทำการคำนวณหาระดับ fair share และทำการทิ้งเซลล์ที่มีการใช้งานบัฟเฟอร์มากเกินไปกว่า fair share กว้างจึงทำให้อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีการ update จำนวนเซลล์ที่อยู่ในสวิทช์มากขึ้น ผลคือสวิทช์จะรู้สถานะของสวิทช์ตลอดเวลาและควรจะทำทิ้งเซลล์ของ VC ไคเพื่อทำให้มีความเท่าเทียมกัน

การเลือกระดับจุดเริ่มเปลี่ยนด้านสูงและด้านต่ำของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีความสำคัญ จะต้องเลือกระดับจุดเริ่มเปลี่ยนด้านสูงก่อน โดยเลือกระดับจุดเริ่มเปลี่ยนด้านสูงที่ไม่ทำให้เกิดปัญหาความไม่สมบูรณ์ของแพ็กเกจ TCP ขึ้นในโครงข่ายทำให้สูญเสียแบนด์วิดท์ เมื่อเลือกระดับจุดเริ่มเปลี่ยนด้านสูงเพื่อให้ได้ประสิทธิภาพสูงสุดแล้วจากนั้นจึงมาทำการเลือกระดับจุดเริ่มเปลี่ยนด้านต่ำ โดยถ้าเลือกให้จุดเริ่มเปลี่ยนด้านต่ำแตกต่างจากจุดเริ่มเปลี่ยนด้านสูงมากก็จะทำให้ค่าความเท่าเทียมกันมากขึ้นด้วย

5.5 ผลการจำลองในกรณีเปลี่ยนค่าต่างๆของ TCP

ในวิทยานิพนธ์นี้ได้ทำการจำลองการทำงานบนแบบจำลองแบบต่างๆ ทั้งในโครงข่ายแบบ LAN และแบบ WAN นอกจากนี้ในวิทยานิพนธ์จะทำการศึกษาถึงตัวแปรของ TCP ที่มีผลกระทบต่อระบบการทำงาน of โครงข่าย ซึ่งตัวแปรของ TCP ที่มีผลกระทบต่อการทำงาน of โครงข่าย เช่น จำนวนแหล่งกำเนิดที่เชื่อมต่อเข้ากับโครงข่าย (จำนวน connection), ขนาด Maximum Transmission Unit (MTU), และค่า Retransmission Time Out (RTO) เป็นต้น ผลการจำลองในส่วนนี้จะกล่าวถึงการเปลี่ยนค่าต่างๆของ TCP บนแบบจำลอง Peer to Peer ในโครงข่ายแบบ LAN โดยแยกพิจารณาทีละตัวแปร ในลำดับแรกจะพิจารณาถึงผลกระทบของจำนวนแหล่งกำเนิดที่เชื่อมต่อเข้ากับโครงข่าย (จำนวน connections) ต่อมาจะศึกษาผลกระทบของขนาด MTU และศึกษาผลกระทบของค่า Retransmission Time Out ที่มีต่อโครงข่าย สุดท้ายจะทำการศึกษาผลการจำลองของกลไก Fast Retransmission ของ TCP

5.5.1 ผลกระทบของจำนวนแหล่งกำเนิด

ผลการจำลองในส่วนนี้จะศึกษาถึงผลกระทบของจำนวนแหล่งกำเนิด จะมีผลต่อสมรรถนะของ TCP อย่างไร และจำนวนแหล่งกำเนิดจะมีผลข้างเคียงกับอะไรบ้าง โดยผลการจำลองนี้จะทำการปรับจำนวนแหล่งกำเนิดเป็น 5,10,15,25,50 และ 100 บัฟเฟอร์มีขนาด 10000 เซลล์ ระดับจุดเริ่มเปลี่ยนของ EPD มีค่า 6000 เซลล์ ส่วนจุดเริ่มเปลี่ยนด้านสูงและด้านต่ำของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีค่า 6000 และ 3000 เซลล์ ตามลำดับ เงื่อนไขต่างๆของแบบจำลอง Peer to Peer เหมือนกับในบทที่ 4

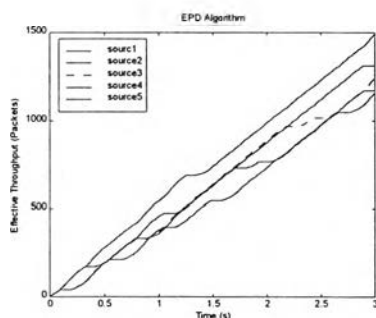
ตารางที่ 5.13 แสดงผลการจำลองการเปลี่ยนจำนวนแหล่งกำเนิดของอัลกอริทึม EPD ตารางที่ 5.14 แสดงผลการจำลองการเปลี่ยนจำนวนแหล่งกำเนิดของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ตารางที่ 5.15 แสดงผลการจำลองการเปลี่ยนจำนวนแหล่งกำเนิดของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต รูปที่ 5.29 แสดงผลการจำลองการเปลี่ยนจำนวนแหล่งกำเนิดของอัลกอริทึม EPD รูปที่ 5.30 แสดงผลการจำลองการเปลี่ยนจำนวนแหล่งกำเนิดของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting รูปที่ 5.31 แสดงผลการจำลองการเปลี่ยนจำนวนแหล่งกำเนิดของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

◆ ผลการจำลองการเปลี่ยนแปลงจำนวนแหล่งกำเนิด ของอัลกอริทึม EPD

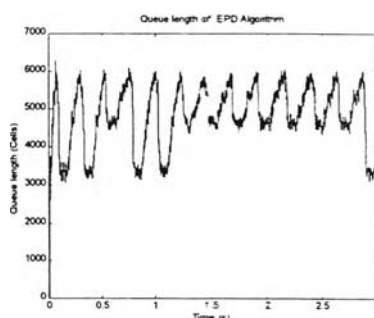
ตารางที่ 5.13 ผลการจำลองการเปลี่ยนแปลงจำนวนแหล่งกำเนิด ของอัลกอริทึม Early Packet Discard

| Number of source | 5 | 10 | 15 | 25 | 50 | 100 |
|-------------------------------|--------|--------|--------|--------|--------|--------|
| Effective throughput (packet) | 6375 | 6290 | 6245 | 6070 | 6057 | 6098 |
| Total Throughput (Mbit/s) | 134.21 | 133.03 | 132.34 | 128.86 | 127.14 | 132.37 |
| Fairness | 0.9866 | 0.9668 | 0.9635 | 0.9577 | 0.9145 | 0.01 |
| Efficiency (%) | 99.15 | 98.27 | 97.76 | 95.19 | 93.92 | 97.78 |

❖ 5 แหล่งกำเนิด

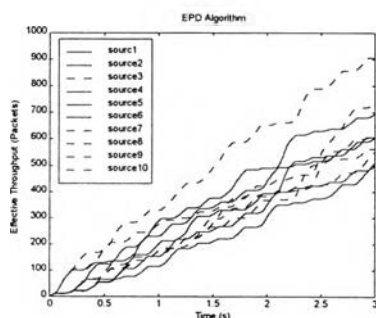


(ก)

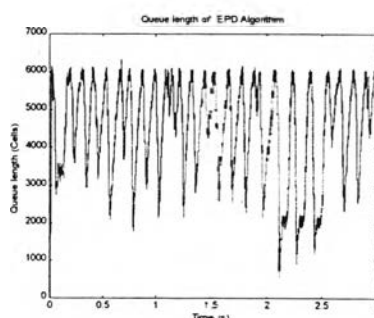


(ข)

❖ 10 แหล่งกำเนิด

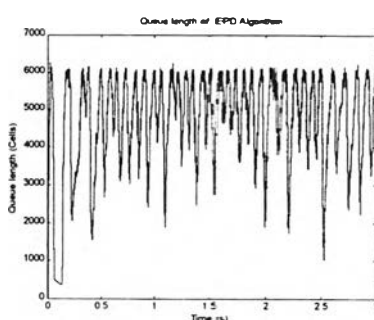
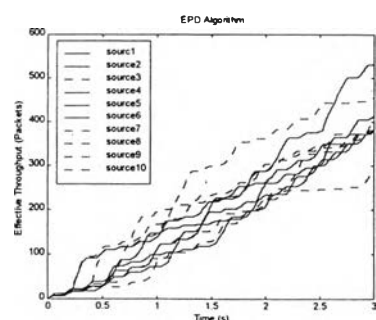


(ค)



(ง)

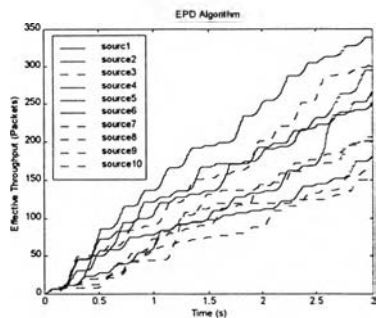
❖ 15 แหล่งกำเนิด



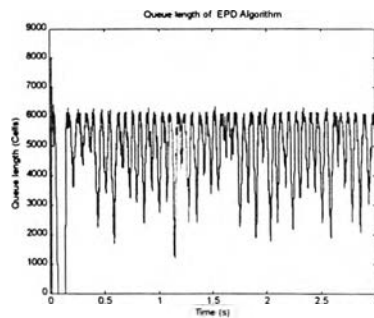
(จ)

(ฉ)

❖ 25 แหล่งกำเนิด

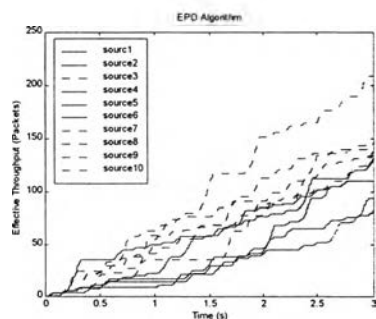


(ช)

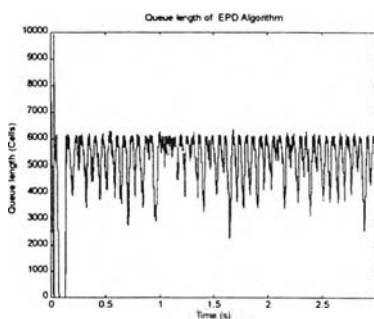


(ซ)

❖ 50 แหล่งกำเนิด

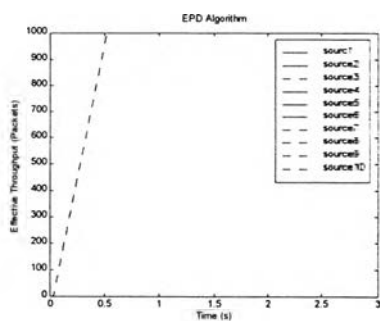


(ฎ)

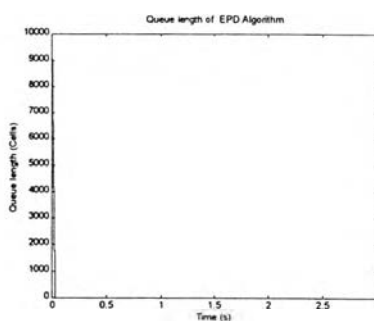


(ฏ)

❖ 100 แหล่งกำเนิด



(ณ)



(ณ)

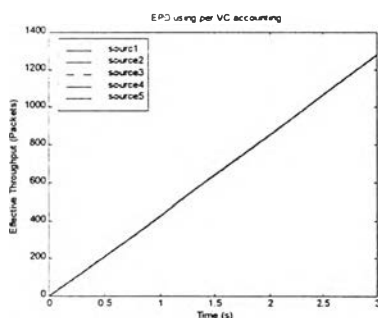
รูปที่ 5.29 ผลการจำลองการเปลี่ยนจำนวนแหล่งกำเนิดของอัลกอริทึม Early Packet Discard

◆ ผลการจำลองการเปลี่ยนแปลงจำนวนแหล่งกำเนิด ของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting

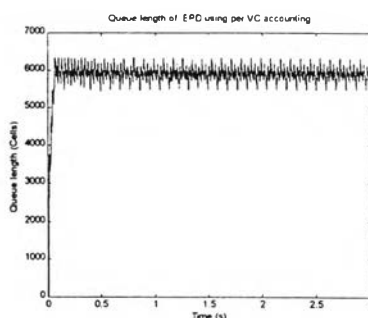
ตารางที่ 5.14 ผลการจำลองการเปลี่ยนแปลงจำนวนแหล่งกำเนิดของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting

| Number of source | 5 | 10 | 15 | 25 | 50 | 100 |
|-------------------------------|--------|--------|--------|--------|--------|--------|
| Effective throughput (packet) | 6392 | 6113 | 5974 | 6015 | 5920 | 6098 |
| Total Throughput (Mbit/s) | 134.39 | 129.31 | 126.02 | 125.87 | 125.34 | 132.37 |
| Fairness | 0.9999 | 0.986 | 0.9809 | 0.9809 | 0.8693 | 0.01 |
| Efficiency (%) | 99.28 | 95.53 | 93.09 | 92.99 | 92.59 | 92.78 |

❖ 5 แหล่งกำเนิด

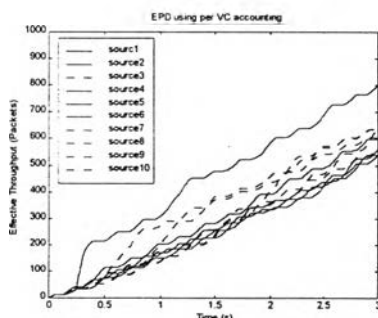


(ก)

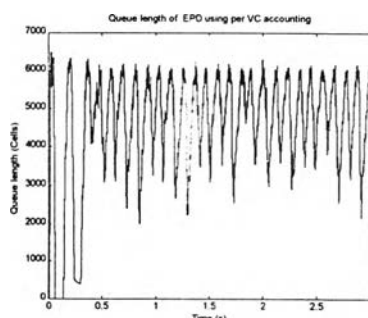


(ข)

❖ 10 แหล่งกำเนิด

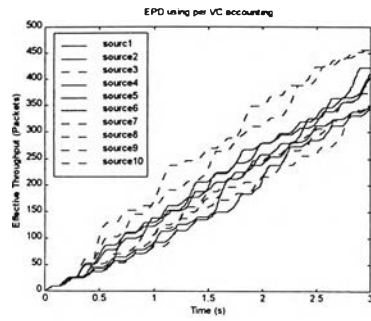


(ค)

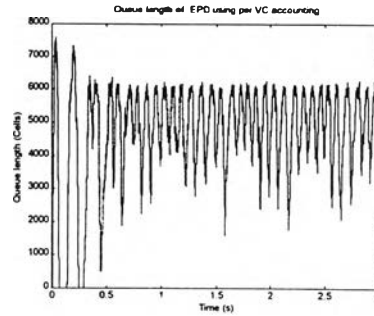


(ง)

❖ 15 แหล่งกำเนิด

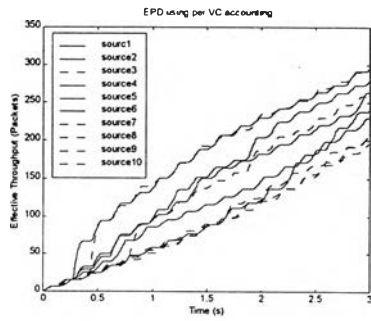


(จ)

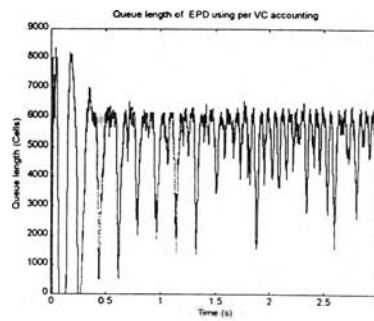


(ฉ)

❖ 25 แหล่งกำเนิด

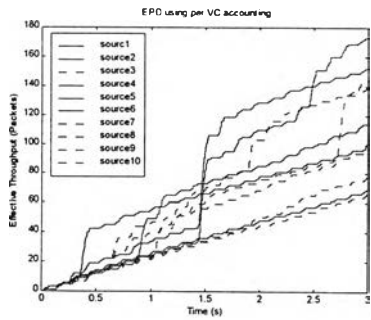


(ง)

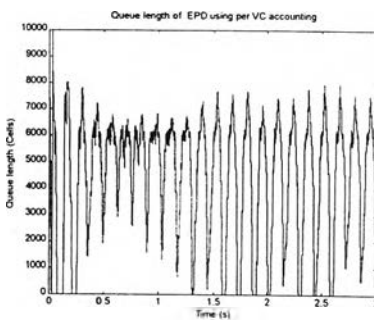


(ช)

❖ 50 แหล่งกำเนิด

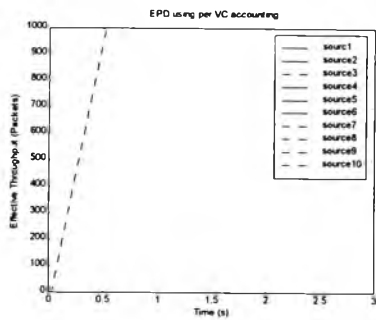


(ค)

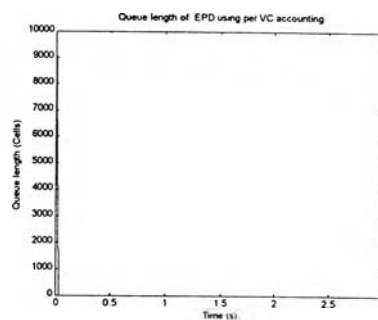


(ง)

❖ 100 แหล่งกำเนิด



(ด)



(ณ)

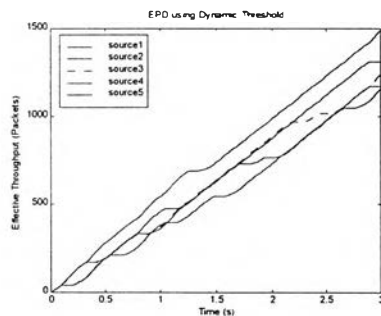
รูปที่ 5.30 ผลการจำลองการเปลี่ยนจำนวนแหล่งกำเนิดของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting

◆ ผลการจำลองการเปลี่ยนแปลงจำนวนแหล่งกำเนิด ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

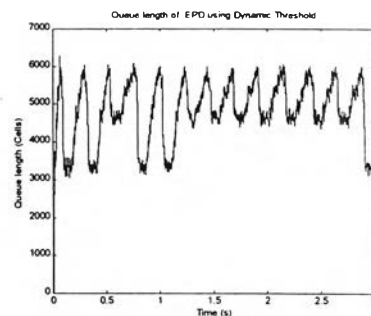
ตารางที่ 5.15 ผลการจำลองการเปลี่ยนจำนวนแหล่งกำเนิด ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

| Number of source | 5 | 10 | 15 | 25 | 50 | 100 |
|-------------------------------|--------|--------|--------|--------|--------|--------|
| Effective throughput (packet) | 6376 | 6109 | 6063 | 6052 | 6056 | 6098 |
| Total Throughput (Mbit/s) | 134.36 | 129.4 | 128.77 | 128.1 | 127.78 | 132.37 |
| Fairness | 0.9936 | 0.9919 | 0.9907 | 0.9925 | 0.9795 | 0.01 |
| Efficiency (%) | 99.26 | 95.59 | 95.12 | 94.69 | 94.4 | 97.79 |

◆ 5 แหล่งกำเนิด

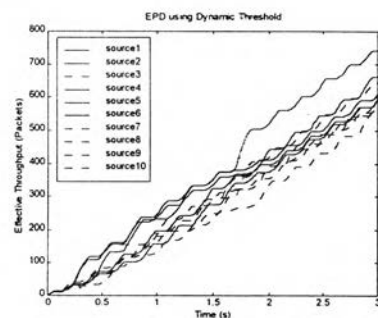


(ก)

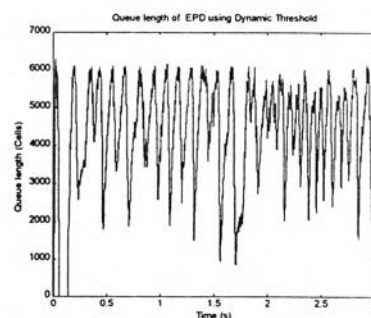


(ข)

◆ 10 แหล่งกำเนิด

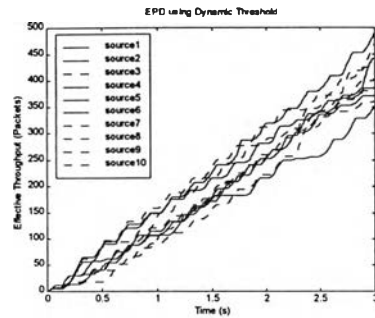


(ค)

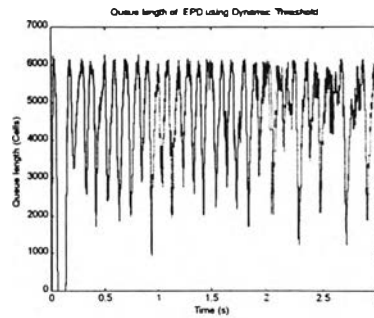


(ง)

❖ 15 แหล่งกำเนิด

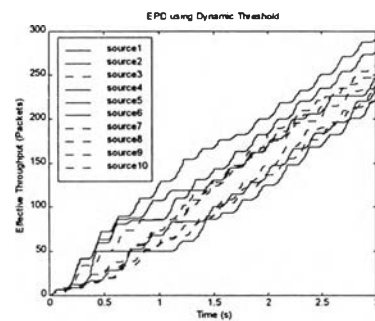


(จ)

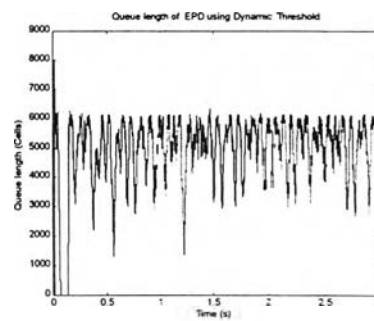


(ฉ)

❖ 25 แหล่งกำเนิด

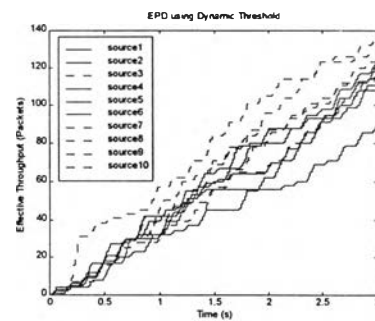


(ช)

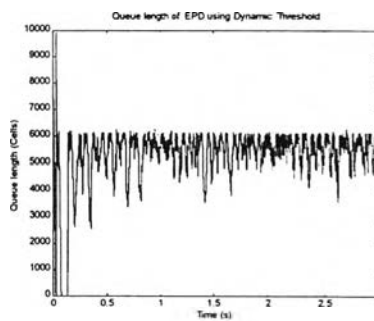


(ซ)

❖ 50 แหล่งกำเนิด

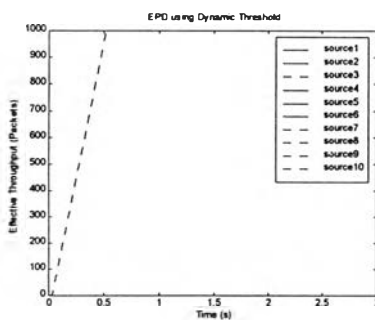


(ค)

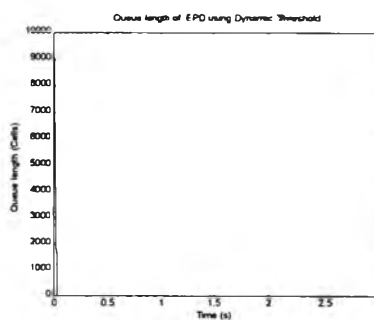


(ก)

❖ 100 แหล่งกำเนิด



(ค)



(ค)

จากผลการจำลองผลกระทบของจำนวนแหล่งกำเนิดหรือจำนวน connection สามารถสรุปได้ดังนี้

- จำนวน connection จะมีผลต่อประสิทธิภาพของ TCP และค่าความเท่าเทียมกัน (fairness) โดยถ้าจำนวน connection มากจะทำให้ค่าประสิทธิภาพ และค่าความเท่าเทียมกันลดลง ส่วนอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต จะยังคงมีค่าประสิทธิภาพของ TCP ลดลงตามจำนวน connection แต่ค่า fairness จะค่อนข้างคงที่โดยจะไม่แปรผันกับจำนวน connection เนื่องจากอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตจะมีการนับจำนวนเซลล์ในบัฟเฟอร์ของสวิตช์ ถ้ามี VC ใดมีจำนวนเซลล์มากกว่า fair share สวิตช์ก็จะทำการทิ้งเซลล์ของ VC นั้นออกไป ซึ่งจะเป็นผลทำให้อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตมีค่า fair share มากขึ้นกว่าอัลกอริทึม EPD
- จากผลการจำลองการเปลี่ยนแปลงจำนวนแหล่งกำเนิดดังในตารางที่ 5.13, 5.14 และ 5.15 จะพบว่าอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting และอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต จะมีค่าประสิทธิภาพของ TCP ที่ต่ำกว่าอัลกอริทึม EPD เป็นเพราะว่าทั้ง 2 อัลกอริทึม จะมีการนับเซลล์ในแต่ละ VC ให้ไม่เกินระดับ fair share ถ้า VC ใดเกินระดับ fair share สวิตช์ก็จะทำการทิ้งเซลล์ที่เข้ามาของ VC นั้นออกไป เมื่อมีการทิ้งเซลล์นี้เกิดขึ้นจึงเป็นผลทำให้ TCP มีประสิทธิภาพลดลง
- จากการจำลองการเปลี่ยนแปลงจำนวนแหล่งกำเนิดหรือจำนวน connection จะพบว่าจำนวน connection จะมีผลกระทบกับค่าต่างๆ ดังนี้

1. การตั้งขนาดจุดเริ่มเปลี่ยน (threshold) โดยจำนวนแหล่งกำเนิดจะมีผลกับขนาดจุดเริ่มเปลี่ยนดังนี้

$$\text{ถ้าค่า } Threshold < MaximumWindow\ size \times connections \left(\frac{65536 \times 5}{53} = 6128 \text{ cells} \right) \text{ จะทำให้}$$

ค่าประสิทธิภาพของ TCP และความเท่าเทียมกันต่ำกว่า 1 (คือมีแฟ็กเกจสูญเสีย)

ถ้าค่า $Threshold > MaximumWindow\ size \times connections$ จะทำให้ค่าประสิทธิภาพของ TCP และค่าความเท่าเทียมกัน มีค่าเป็น 1 (คือไม่มีการสูญเสีย)

2. ขนาดของบัฟเฟอร์ จากการจำลองพบว่า จำนวน connection จะมีผลกับขนาดบัฟเฟอร์ของสวิตช์ ดังนี้

ขนาด $Buffer\ size > MTU \times connections$ ($171 \times 100 = 17100 \text{ cells}$) จะมีผลทำให้ fairness index มีค่าสูง

ขนาด $Buffer\ size < MTU \times connections$ จะทำให้ fair share ลดลงอย่างมาก (ในกรณีมี 100 connections จะได้ค่าความเท่าเทียมกันเป็น 0.01) เพราะมี VC เดียวเท่านั้นที่สามารถส่งข้อมูลได้ ส่วน VC อื่นจะถูกทิ้งทั้งหมดเนื่องจากเซลล์มีมากเกินไปที่สวิตช์จะรับได้

ฉะนั้นขนาด บัฟเฟอร์ของสวิตช์อย่างน้อยที่สุดจะต้องมากกว่าขนาด 1 แฟ็กเกจของ TCP จึงจะทำให้มีค่าประสิทธิภาพของ TCP และค่าความเท่าเทียมกันดีขึ้น

5.5.2 ผลการจำลองสมรรถนะ TCP เมื่อมีการเปลี่ยนแปลงขนาด MTU (Maximum Transmission Unit)

ในการจำลองส่วนนี้จะทำการเปลี่ยนแปลงขนาด MTU ของโพรโตคอล TCP เพื่อศึกษาการเปลี่ยนแปลงขนาด MTU จะมีผลอย่างไรเกี่ยวกับอัลกอริทึมแบบต่างที่เสนอขึ้นมาและสมรรถนะของ TCP ในกรณีของความเท่าเทียมกัน (fairness) และค่าวิสัยความสามารถ (throughput) ค่า MTU ที่ได้เสนอขึ้นมาทำการจำลองมีค่าดังต่อไปนี้ 1500, 4352, 8192 และ 9180 ไบต์ ซึ่งค่าที่ได้เสนอขึ้นมาทำการจำลองเป็นขนาดของแพ็กเกจของ TCP ที่ถูกนำมาใช้ในปัจจุบันคือขนาด 1500 ไบต์ เป็นขนาดของแพ็กเกจที่ใช้ในโครงข่าย ethernet ขนาด 4352 ไบต์ คือขนาดแพ็กเกจของโครงข่าย FDDI ขนาด 8192 ไบต์คือขนาดของ file transfer และ 9180 คือขนาดแพ็กเกจที่ใช้สำหรับ IP ในโครงข่าย ATM

จากผลการจำลองนี้จะมีการเปลี่ยนขนาด MTU ซึ่งขนาดของ MTU เท่ากับขนาดแพ็กเกจของ TCP รวมกับส่วนหัวของ TCP และส่วนหัวของ IP เมื่อมีการเปลี่ยนขนาด MTU ฉะนั้นจะต้องมีการหาประสิทธิภาพของ TCP ใหม่ทุกครั้งที่มีการเปลี่ยนขนาด MTU วิธีการหาประสิทธิภาพและสมรรถนะของ TCP ดังแสดงในส่วนที่ 4.2 โดยขนาดแพ็กเกจ TCP เป็น 1500 ไบต์ จะมีค่าวิสัยสามารถที่เป็นไปได้ที่มากที่สุดเท่ากับ 132.4 Mbps เมื่อขนาดแพ็กเกจของ TCP เป็น 4352 ไบต์ จะมีค่าวิสัยสามารถที่เป็นไปได้ที่มากที่สุดเท่ากับ 135.13 Mbps ส่วนขนาดแพ็กเกจของ TCP เป็น 8192 ไบต์ จะมีค่าวิสัยสามารถที่เป็นไปได้มากที่สุดเท่ากับ 135.37 Mbps และเมื่อขนาดแพ็กเกจของ TCP เป็น 9180 ไบต์ จะมีค่าวิสัยสามารถที่เป็นไปได้ที่มากที่สุดเท่ากับ 135.1 Mbps

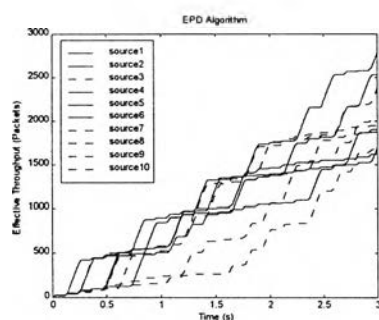
ข้อกำหนดของแบบจำลองคือ ขนาดของบัฟเฟอร์คือ 3000 เซลล์, TCP window size ที่ใช้เป็น 64 kB, retransmission time out คือ 100 msec, โครงข่ายที่ใช้ทำการจำลองเป็นแบบ peer to peer ที่มี 10 แหล่งกำเนิดต่อเข้ากับสวิตช์ โดยมีระยะห่างระหว่างแหล่งกำเนิดกับสวิตช์เป็นระยะทาง 600 เมตร เวลาที่ใช้ในการจำลองคือ 3 นาที ส่วนกำหนดอื่นๆจะเหมือนกับส่วนที่ 4.1.1

ผลการจำลองการเปลี่ยนขนาดของ MTU ของแต่ละอัลกอริทึมของสวิตช์แสดงในตารางที่ 5.16-5.18 โดยตารางที่ 5.16 แสดงผลการจำลองการเปลี่ยนขนาด MTU ของอัลกอริทึม EPD ตารางที่ 5.17 แสดงผลการจำลองการเปลี่ยนขนาด MTU ของอัลกอริทึม EPD-per-VC accounting ส่วนตารางที่ 5.18 แสดงผลการจำลองการเปลี่ยนขนาด MTU ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนแบบพลวัต ส่วนในรูปที่ 5.32 ถึง 5.35 แสดงค่า Effective Throughput ของแต่ละแหล่งกำเนิดซึ่งค่า Effective Throughput สามารถแสดงถึงความเท่าเทียมกันได้ และผลของความยาวคิวที่สวิตช์แบบ EPD ในรูปที่ 5.36 ถึง 5.39 แสดงผลของ Effective Throughput และผลของความยาวคิวที่สวิตช์เป็นแบบ EPD ที่ใช้เทคนิค per-VC accounting รูปที่ 5.40 ถึง 5.43 แสดงผลของ Effective Throughput และความยาวคิวที่สวิตช์เป็นแบบ EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

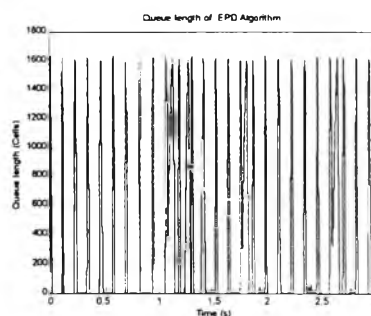
❖ ผลการจำลองของอัลกอริทึม EPD

ตารางที่ 5.16 ผลการจำลองการเปลี่ยนขนาด MTU ของอัลกอริทึม Early Packet Discard

| MTU | 1540 | 4392 | 8232 | 9220 |
|-------------------------------|--------|--------|--------|--------|
| Effective Throughput (packet) | 20425 | 9830 | 5366 | 4925 |
| Total TCP throughput (Mbit/s) | 77.276 | 110.74 | 113.74 | 116.88 |
| Total Loss (cells) | 18592 | 34799 | 56522 | 60096 |
| Fairness | 0.9672 | 0.9567 | 0.9379 | 0.9479 |
| Efficiency (%) | 58.34 | 81.95 | 84.02 | 86.52 |

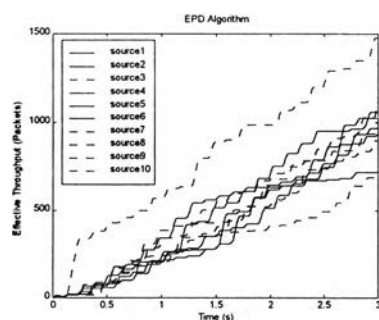


(ก)

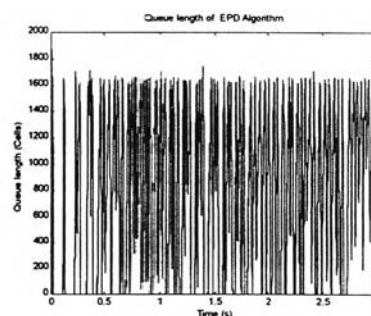


(ข)

รูปที่ 5.32 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD
เมื่อขนาด MTU = 1500 bytes

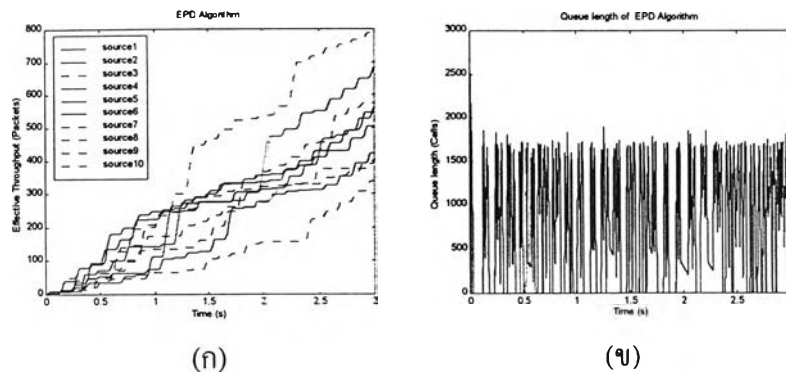


(ก)

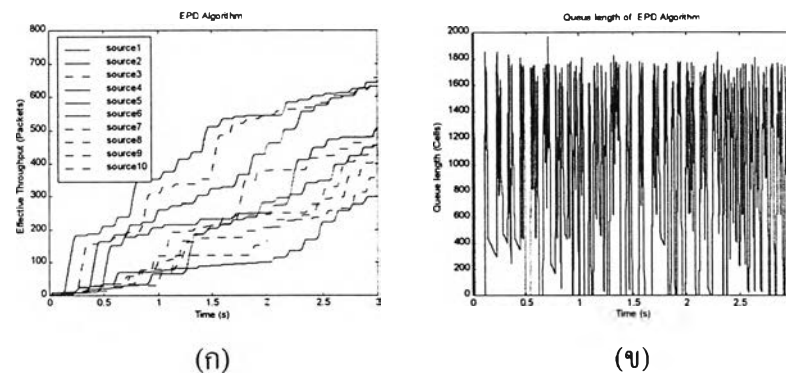


(ข)

รูปที่ 5.33 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD
เมื่อขนาด MTU = 4352 bytes



รูปที่ 5.34 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD
เมื่อขนาด MTU = 8192 bytes

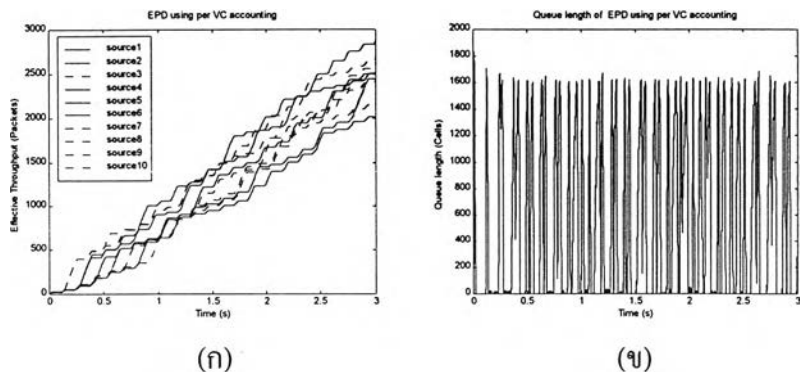


รูปที่ 5.35 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD
เมื่อขนาด MTU = 9180 bytes

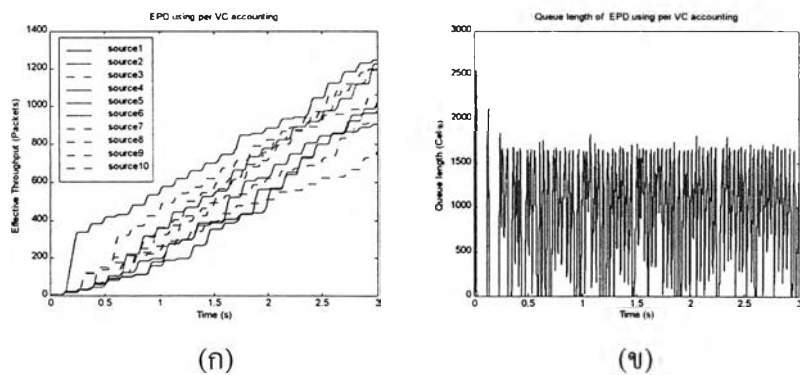
❖ ผลการจำลองของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting

ตารางที่ 5.17 ผลการจำลองการเปลี่ยนขนาด MTU ของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting

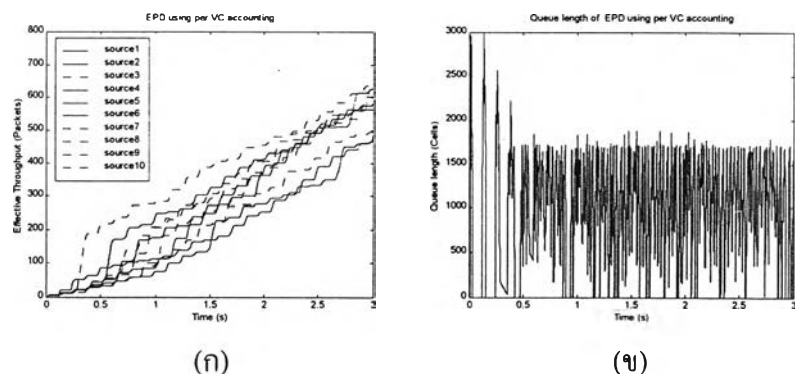
| MTU | 1500 | 4092 | 8232 | 9220 |
|-------------------------------|--------|--------|--------|--------|
| Effective Throughput (packet) | 24694 | 10537 | 5655 | 4841 |
| Total TCP throughput (Mbit/s) | 93.662 | 118.29 | 119.46 | 114.37 |
| Total Loss (cells) | 17664 | 39275 | 65062 | 67375 |
| Fairness | 0.9916 | 0.9791 | 0.9891 | 0.9664 |
| Efficiency (%) | 70.71 | 87.53 | 88.25 | 84.65 |



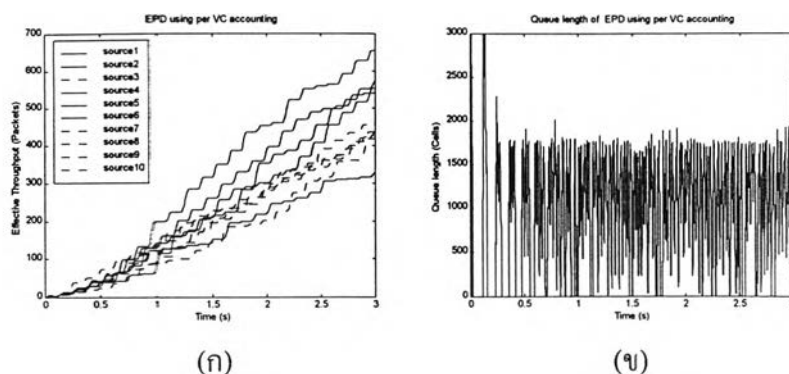
รูปที่ 5.36 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค per-VC accounting เมื่อขนาด MTU = 1500 bytes



รูปที่ 5.37 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค per-VC accounting เมื่อขนาด MTU = 4352 bytes



รูปที่ 5.38 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค per-VC accounting เมื่อขนาด MTU = 8192 bytes

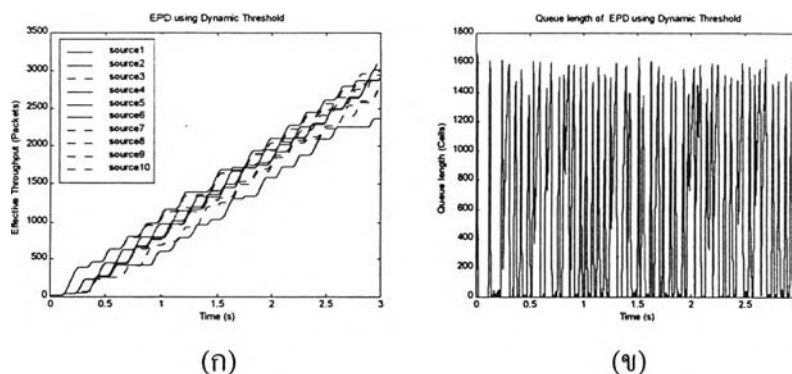


รูปที่ 5.39 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค per-VC accounting เมื่อขนาด MTU = 9180 bytes

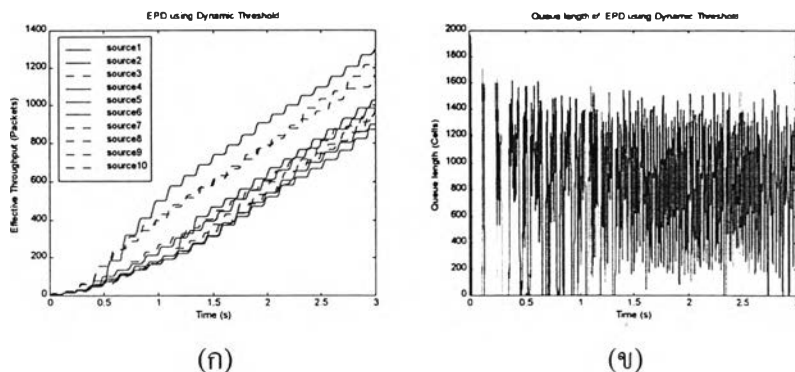
❖ ผลการจำลองของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

ตารางที่ 5.18 ผลการจำลองการเปลี่ยนขนาด MTU ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

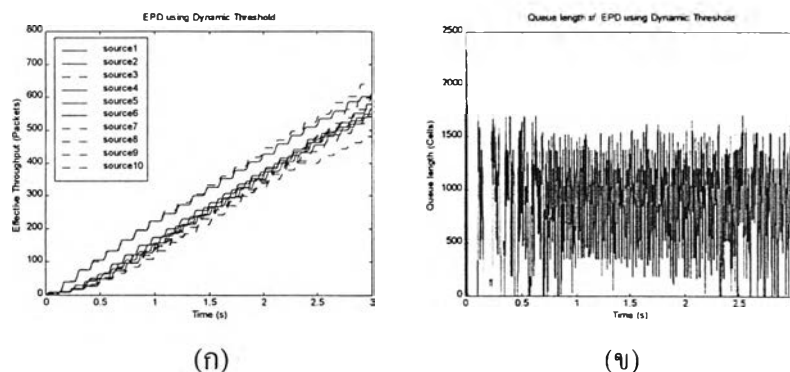
| MTU | 1540 | 4392 | 8232 | 9220 |
|--------------------------------|--------|--------|--------|--------|
| Effective Throughput (packets) | 28775 | 10335 | 5727 | 5065 |
| Total TCP throughput (Mbit/s) | 108.37 | 115.32 | 119.5 | 119.28 |
| Total Loss (cells) | 24768 | 49838 | 92856 | 88018 |
| Fairness | 0.9978 | 0.9827 | 0.9938 | 0.9621 |
| Efficiency (%) | 81.82 | 85.34 | 88.27 | 88.29 |



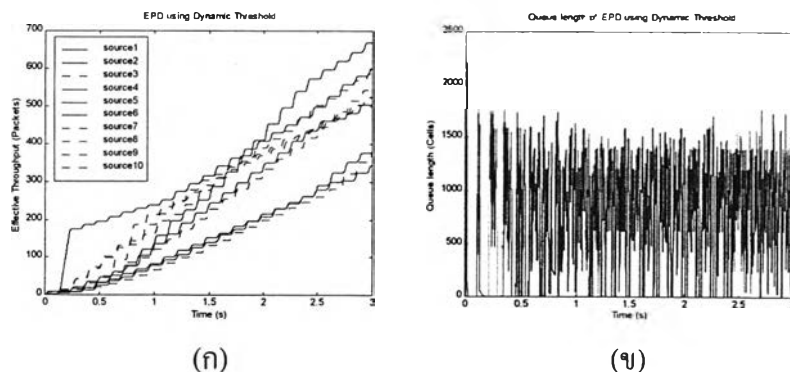
รูปที่ 5.40 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต เมื่อขนาด MTU = 1500 bytes



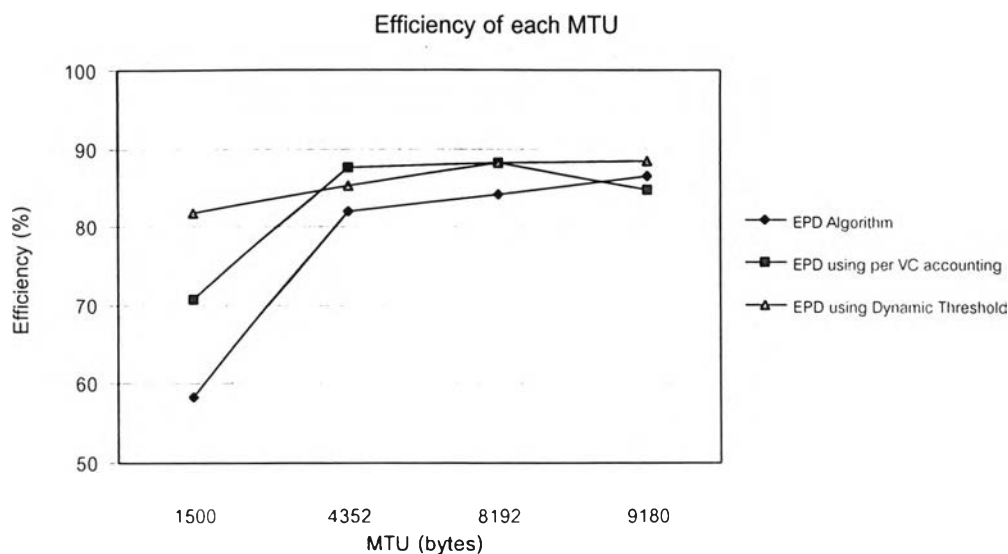
รูปที่ 5.41 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค จุดเริ่มเปลี่ยนพลวัต เมื่อขนาด MTU = 4352 bytes



รูปที่ 5.42 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิคจุด เริ่มเปลี่ยนพลวัต เมื่อขนาด MTU = 8192 bytes



รูปที่ 5.43 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค จุดเริ่มเปลี่ยนพลวัต เมื่อขนาด MTU = 9180 bytes



รูปที่ 5.44 แสดงประสิทธิภาพของโครงข่ายของแต่ละอัลกอริทึม เมื่อมีการเปลี่ยนขนาด MTU

การวิเคราะห์ผลการจำลองการเปลี่ยนขนาด MTU

จากตารางที่ 5.16 ถึง 5.18 จะพบว่าขนาด MTU มากขึ้นก็จะมีผลทำให้ประสิทธิภาพของ TCP มากขึ้นด้วย ยกเว้นขนาด MTU เป็น 9180 bytes ของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting โดยมีประสิทธิภาพลดลง 3.6% แทนที่จะเพิ่มขึ้นจากเดิม เพราะเกิดการล้นที่บัฟเฟอร์ทำให้เกิดแพ็กเกจไม่สมบูรณ์ขึ้นในโครงข่าย ดังในรูป 5.39 (ข) เมื่อเกิดแพ็กเกจไม่สมบูรณ์ขึ้นในโครงข่ายจะมีผลทำให้ค่าวิสัยความสามารถและค่าความเท่าเทียมกันลดลง

สาเหตุที่เมื่อขนาด MTU ต่ำจะมีประสิทธิภาพและวิสัยความสามารถต่ำเป็นเพราะมีการใช้แบนด์วิดท์ของโครงข่ายต่ำ ดูจากรูปที่ 5.32, 5.36 และ 5.40 (ข) จึงมีการช่วงเวลาที่ไม่มีข้อมูลจากสวิตช์ถูกส่งเข้าไปในโครงข่าย แต่เมื่อเปรียบเทียบในแต่ละอัลกอริทึมที่สวิตช์แล้วจะพบว่าเมื่อขนาด MTU ต่ำ อัลกอริทึม EPD จะมีการใช้แบนด์วิดท์ของโครงข่ายต่ำที่สุด (ดูจากรูปที่ 5.32 (ข)) จากรูปที่ 5.44 จะสังเกตเห็นว่าที่ขนาด MTU = 4352 bytes อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนแบบพลวัตจะมีประสิทธิภาพของ TCP ต่ำกว่าอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ประมาณ 2.19% เนื่องมาจากในเทคนิคจุดเริ่มเปลี่ยนแบบพลวัตจะใช้จุดเริ่มเปลี่ยน fair share ในการทิ้งเซลล์ต่างๆออกไปเพื่อปรับให้แต่ละ VC มีการส่งข้อมูลได้เท่าเทียมกัน ซึ่งจะพบว่าที่ขนาด MTU=4352 bytes นี้ค่าความเท่าเทียมกันของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนแบบพลวัตมีค่ามากกว่าทุกๆอัลกอริทึม แต่เนื่องจากการทิ้งเซลล์ที่สวิตช์เพราะใช้จุดเริ่มเปลี่ยน fair share เพื่อทำให้ค่าความเท่าเทียมกันดีขึ้นนี้จะมีผลทำให้ค่าวิสัยความสามารถและประสิทธิภาพของ TCP ลดลงด้วย

สาเหตุเมื่อขนาด MTU มากขึ้นแล้วทำให้ประสิทธิภาพของ TCP มากขึ้นด้วยเป็นเพราะ เมื่อ MTU ต่ำ จำนวนเซกเมนต์ต่อหนึ่งแพ็กเกจก็น้อยลง เมื่อความยาวคิวถึงจุดเริ่มเปลี่ยนแล้วสวิตช์จะทำการทิ้งเซกเมนต์ออกไปโดยสวิตช์จะทิ้งเซกเมนต์ทั้งแพ็กเกจออกไปแต่การทิ้งเซกเมนต์ทั้งแพ็กเกจเป็นจำนวนมากถึงจะมีผลทำให้ความยาวคิวที่สวิตช์ลดต่ำกว่าจุดเริ่มเปลี่ยน ฉะนั้นสวิตช์จึงทิ้งเซกเมนต์เพียงหนึ่งแพ็กเกจไม่เพียงพอเพราะหนึ่งแพ็กเกจมีจำนวนเซกเมนต์น้อย สวิตช์จึงต้องทิ้งหลายๆแพ็กเกจจึงเป็นผลทำให้มีการใช้แบนด์วิดท์ของโครงข่ายต่ำลง แต่ถ้าขนาด MTU มาก ฉะนั้นจำนวนเซกเมนต์ต่อหนึ่งแพ็กเกจมีจำนวนมาก เมื่อความยาวคิวมากเกินจุดเริ่มเปลี่ยนสวิตช์ทำการทิ้งเซกเมนต์เป็นจำนวนที่น้อยกว่า ก็เพียงพอที่ทำให้ความยาวคิวต่ำกว่าจุดเริ่มเปลี่ยน จึงเป็นผลทำให้มีการใช้แบนด์วิดท์ที่ดีขึ้น

5.5.3 ผลการจำลองการเปลี่ยนแปลงค่า Retransmission Time Out (RTO)

ในการจำลองส่วนนี้จะทำการเปลี่ยนแปลงค่า Retransmission time out ของโพรโตคอล TCP เพื่อทำการศึกษาการเปลี่ยนแปลงของค่า Retransmission time out จะมีผลอย่างไรกับสมรรถนะของ TCP ในกรณีของวิสัยสามารถและความเท่าเทียมกัน เมื่อใช้อัลกอริทึมแบบต่างๆที่สวิตช์

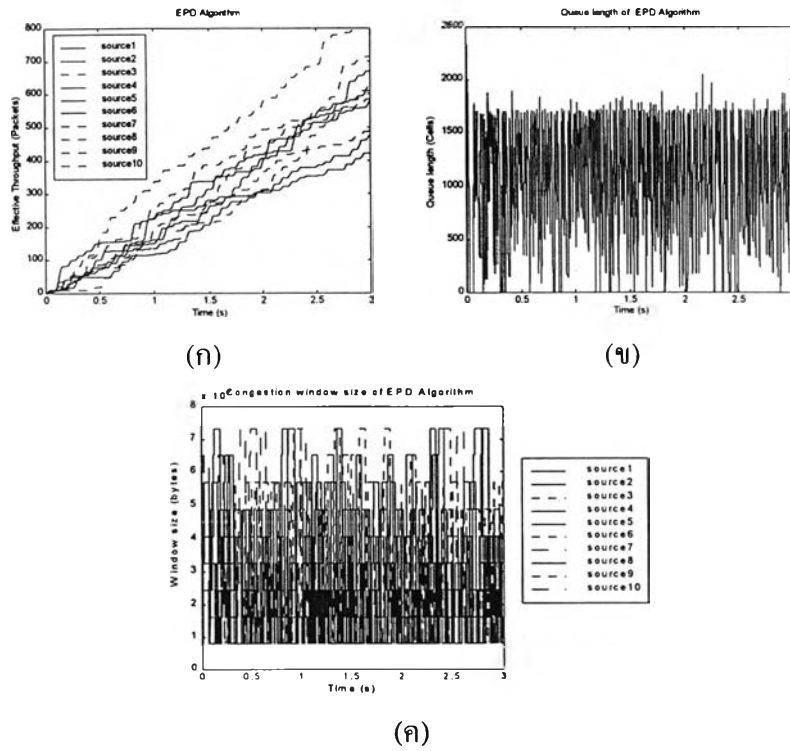
ข้อกำหนดของแบบจำลองคือ ขนาดบัฟเฟอร์ที่สวิตช์เป็น 3000 ขนาดแพ็กเกจของ TCP มีค่าเป็น 8 kB แบบจำลองที่ใช้ในการทดสอบเป็นแบบ Peer to Peer มีจำนวนแหล่งกำเนิดทั้งหมด 10 แหล่งกำเนิด โดยมีระยะห่างระหว่างสวิตช์กับสวิตช์ และสวิตช์กับต้นทางหรือปลายทางเป็น 600 เมตร ซึ่งเป็นโครงข่ายแบบ LAN และเวลาที่ใช้ในการจำลองคือ 3 นาที ส่วนข้อกำหนดอื่นจะเหมือนกับข้อกำหนดของแบบจำลอง Peer to Peer ในส่วนที่ 4.1.1

ผลการจำลองการเปลี่ยนค่า RTO ของแต่ละอัลกอริทึมที่สวิตช์ถูกแสดงในตารางที่ 5.19-5.21 โดยตารางที่ 5.19 แสดงผลการจำลองการเปลี่ยนค่า RTO ของอัลกอริทึม EPD ตารางที่ 5.20 แสดงผลการจำลองการเปลี่ยนค่า RTO ของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting และตารางที่ 5.21 แสดงผลการจำลองการเปลี่ยนค่า RTO ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต รูปที่ 5.45 ถึง 5.48 เป็นผลการจำลองการเปลี่ยนค่า RTO ของสวิตช์ EPD โดยที่ (ก)แสดง Effective Throughput ของแต่ละแหล่งกำเนิด (ข) แสดงความยาวคิวของสวิตช์ (ค) แสดง Congestion window size รูปที่ 5.49 ถึง 5.52 เป็นผลการจำลองการเปลี่ยนค่า RTO ของสวิตช์ EPD ที่ใช้เทคนิค per-VC accounting โดยที่ (ก)แสดง Effective Throughput ของแต่ละแหล่งกำเนิด (ข) แสดงความยาวคิวของสวิตช์ (ค) แสดง Congestion window size และรูปที่ 5.53 ถึง 5.56 เป็นผลการจำลองการเปลี่ยนค่า RTO ของสวิตช์ EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต โดยที่ (ก)แสดง Effective Throughput ของแต่ละแหล่งกำเนิด (ข) แสดงความยาวคิวของสวิตช์ (ค) แสดง Congestion window size

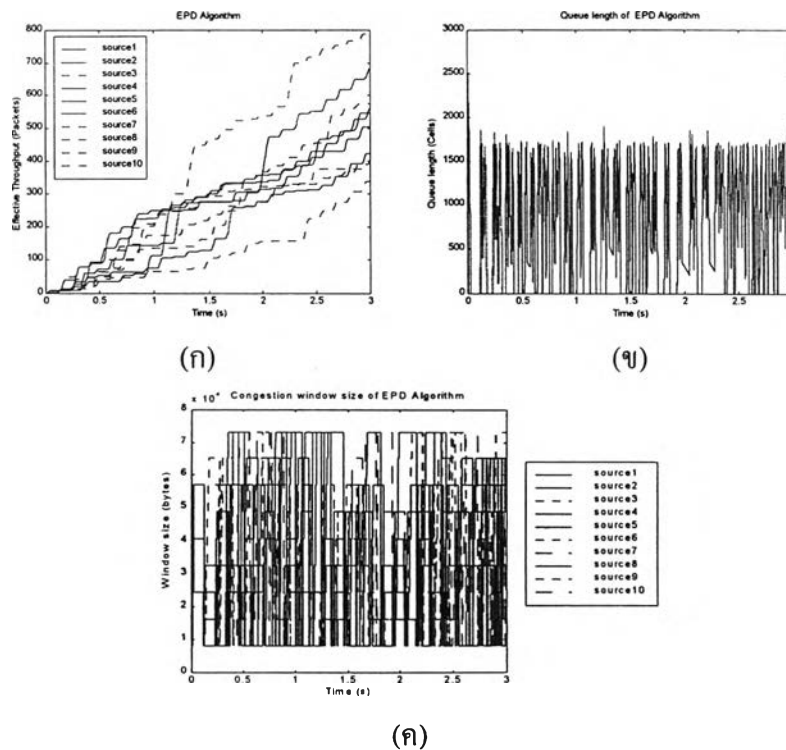
❖ ผลการจำลองของอัลกอริทึม EPD

ตารางที่ 5.19 ผลการจำลองการเปลี่ยนค่า Retransmission Time Out ของอัลกอริทึม Early Packet Discard

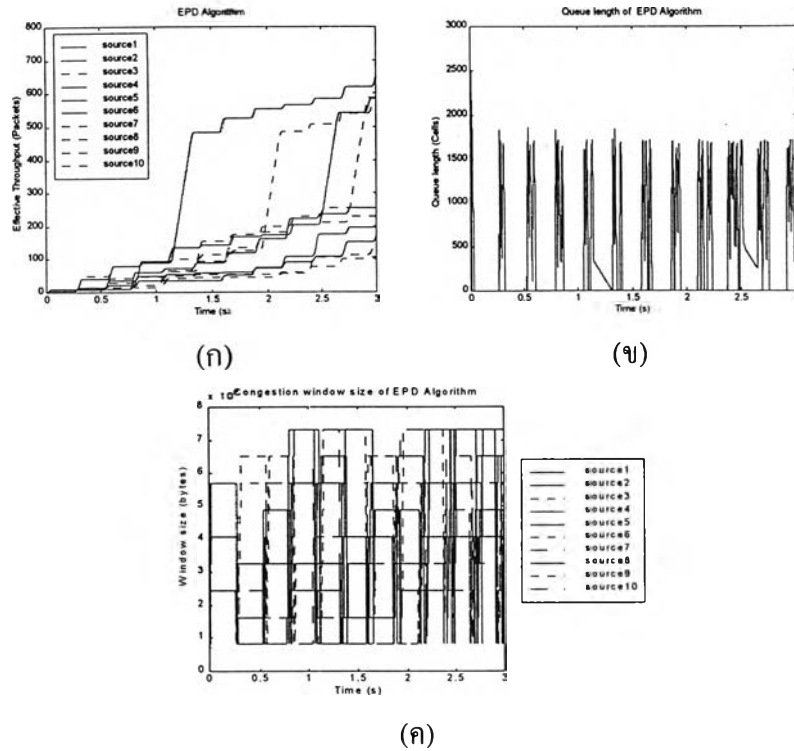
| RTO (msec) | 50 | 100 | 250 | 500 |
|-------------------------------|--------|--------|--------|--------|
| Effective Throughput (packet) | 6007 | 5366 | 3497 | 2489 |
| Total TCP throughput (Mbit/s) | 127.08 | 113.74 | 74.336 | 52.977 |
| Total Loss (cells) | 101364 | 54461 | 27477 | 15629 |
| Fairness | 0.965 | 0.9379 | 0.7411 | 0.3576 |
| Efficiency (%) | 93.87 | 84.02 | 54.91 | 39.13 |



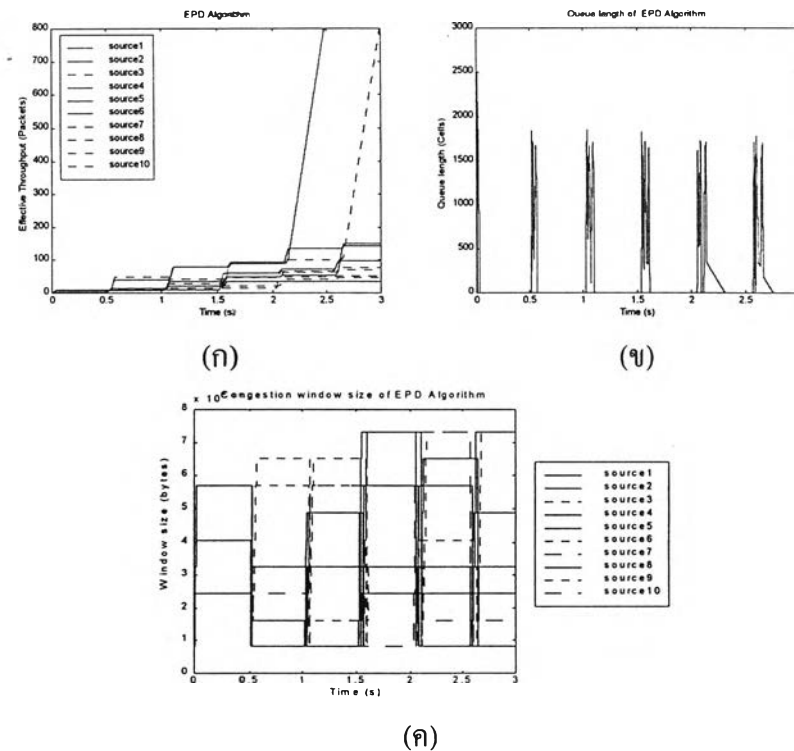
รูปที่ 5.45 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 50 ms



รูปที่ 5.46 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 100 ms



รูปที่ 5.47 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 250 ms



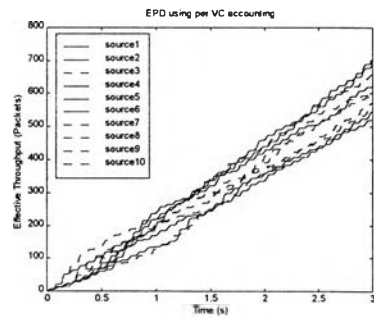
รูปที่ 5.48 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 500 ms

❖ ผลการจำลองของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting

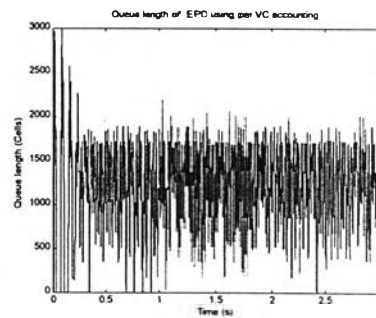
ตารางที่ 5.20 ผลการจำลองการเปลี่ยนค่า Retransmission Time Out ของอัลกอริทึม EPD ที่ใช้เทคนิค

per-VC accounting

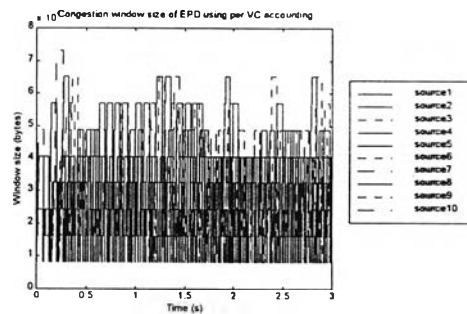
| RTO (msec) | 50 | 100 | 250 | 500 |
|-------------------------------|--------|--------|--------|--------|
| Effective Throughput (packet) | 6058 | 5655 | 3221 | 1456 |
| Total TCP throughput (Mbit/s) | 126.93 | 119.46 | 68.074 | 30.611 |
| Total Loss (cells) | 120908 | 65062 | 28746 | 15074 |
| Fairness | 0.9899 | 0.9891 | 0.7772 | 0.333 |
| Efficiency (%) | 93.77 | 88.25 | 50.29 | 22.61 |



(ก)

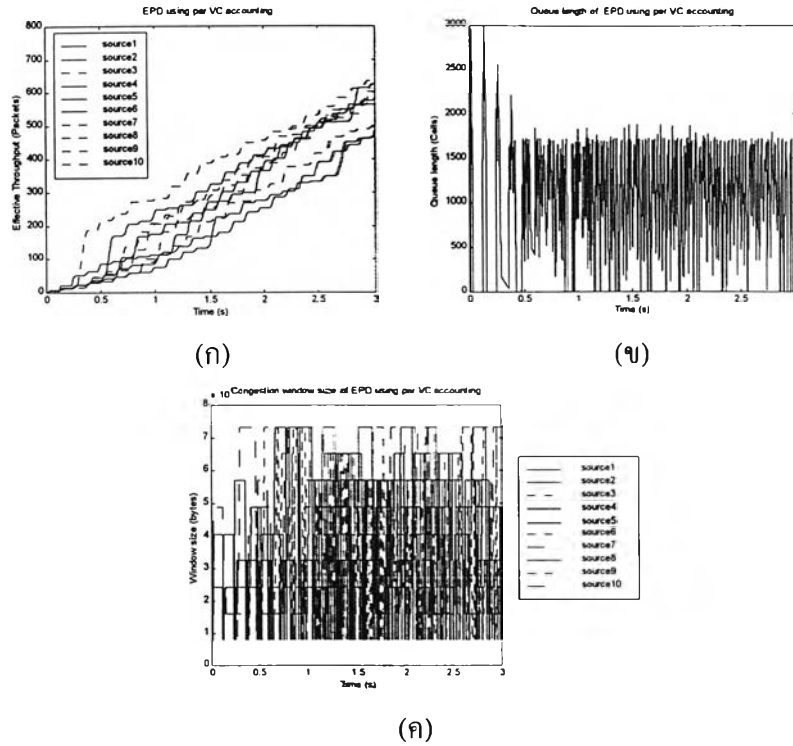


(ข)

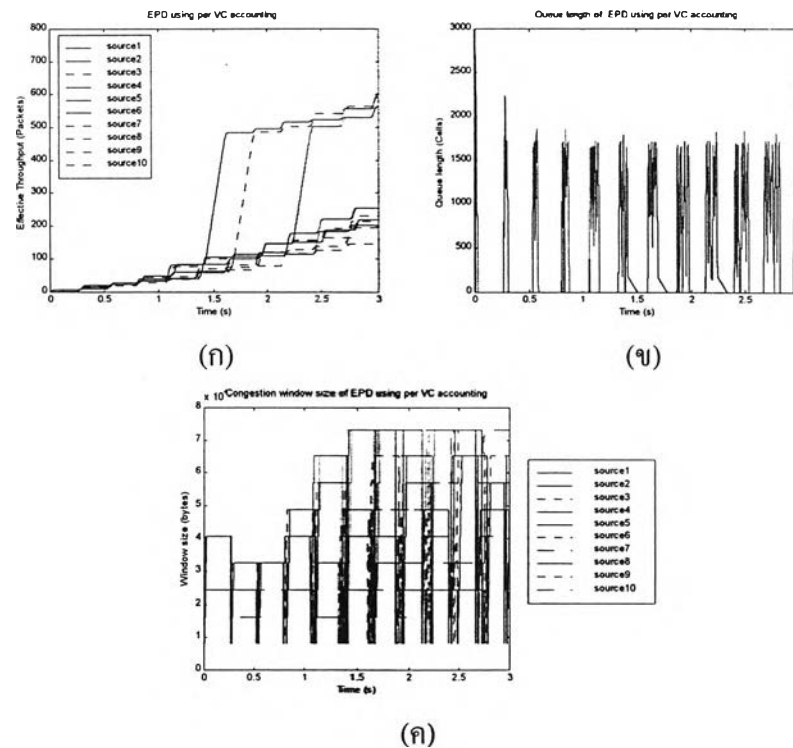


(ค)

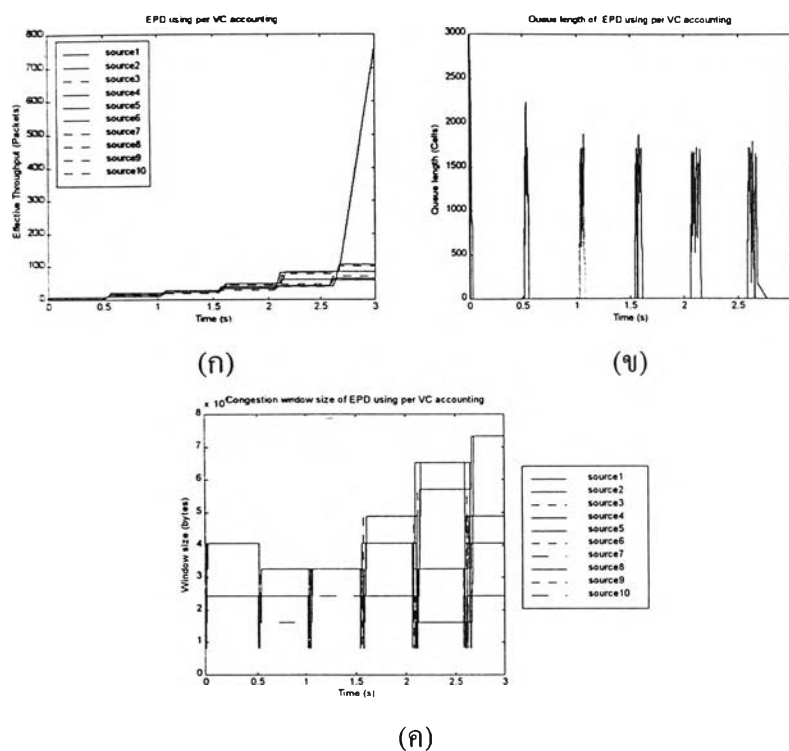
รูปที่ 5.49 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค per-VC accounting และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 50 ms



รูปที่ 5.50 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค per-VC accounting และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 100 ms



รูปที่ 5.51 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค per-VC accounting และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 250 ms

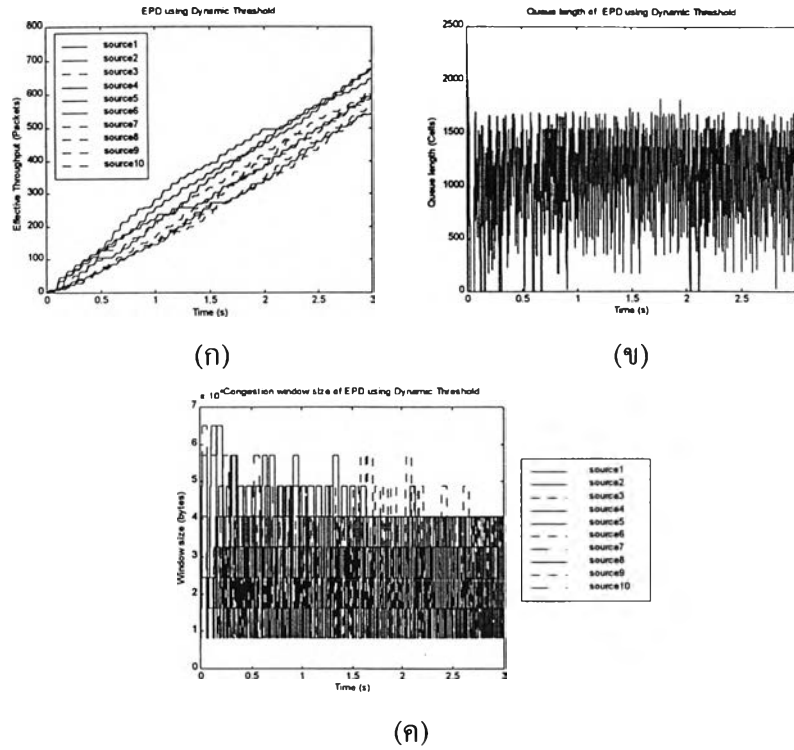


รูปที่ 5.52 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิค per-VC accounting และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 500 ms

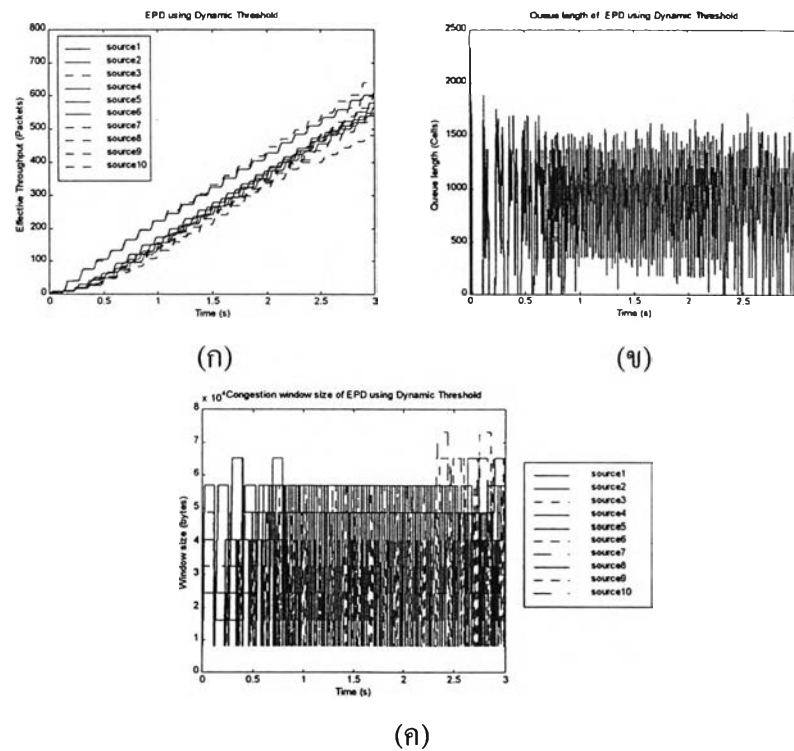
❖ ผลการจำลองของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

ตารางที่ 5.21 ผลการจำลองการเปลี่ยนค่า Retransmission Time Out ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนแบบพลวัต

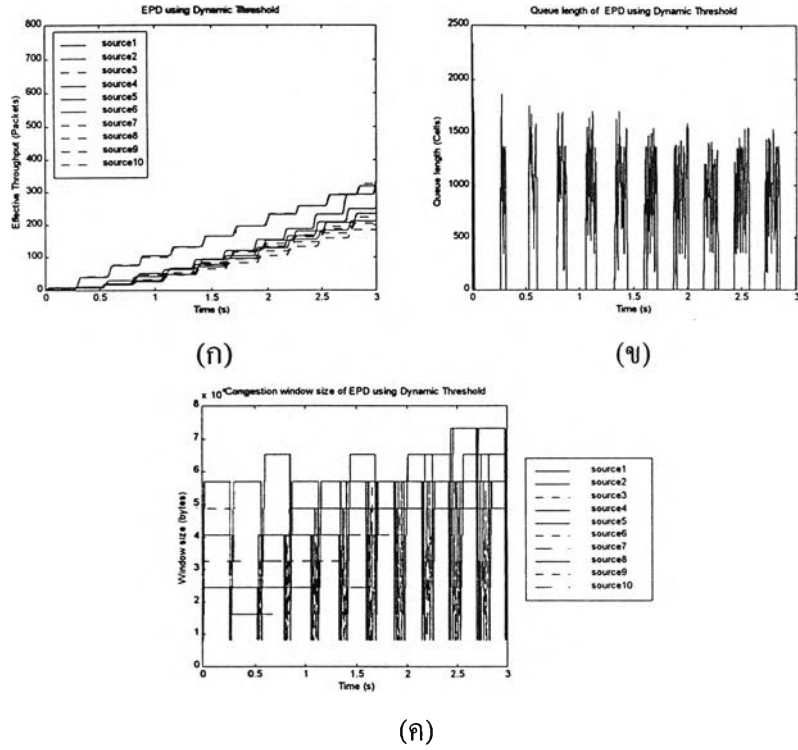
| RTO (msec) | 50 | 100 | 250 | 500 |
|-------------------------------|--------|--------|--------|--------|
| Effective Throughput (packet) | 6088 | 5727 | 2504 | 986 |
| Total TCP throughput (Mbit/s) | 128.29 | 119.5 | 52.214 | 20.064 |
| Total Loss (cells) | 123789 | 92856 | 38657 | 19135 |
| Fairness | 0.9948 | 0.9938 | 0.9562 | 0.8868 |
| Efficiency (%) | 94.77 | 88.27 | 38.57 | 14.82 |



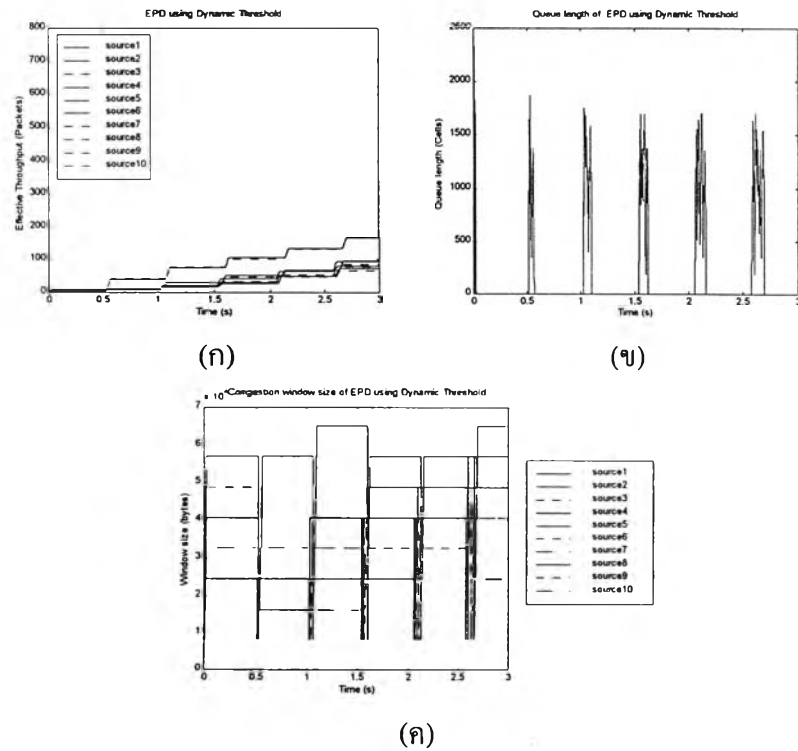
รูปที่ 5.53 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิคจุดเริ่มต้นเปลี่ยนพลวัต และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 50 ms



รูปที่ 5.54 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิคจุดเริ่มต้นเปลี่ยนพลวัต และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 100 ms



รูปที่ 5.55 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 250 ms



รูปที่ 5.56 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวที่สวิตช์ EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต และ (ค) ขนาดของ Congestion Window เมื่อค่า Retransmission Time Out = 500 ms

การวิเคราะห์ผลการเปลี่ยนค่า Retransmission Time Out

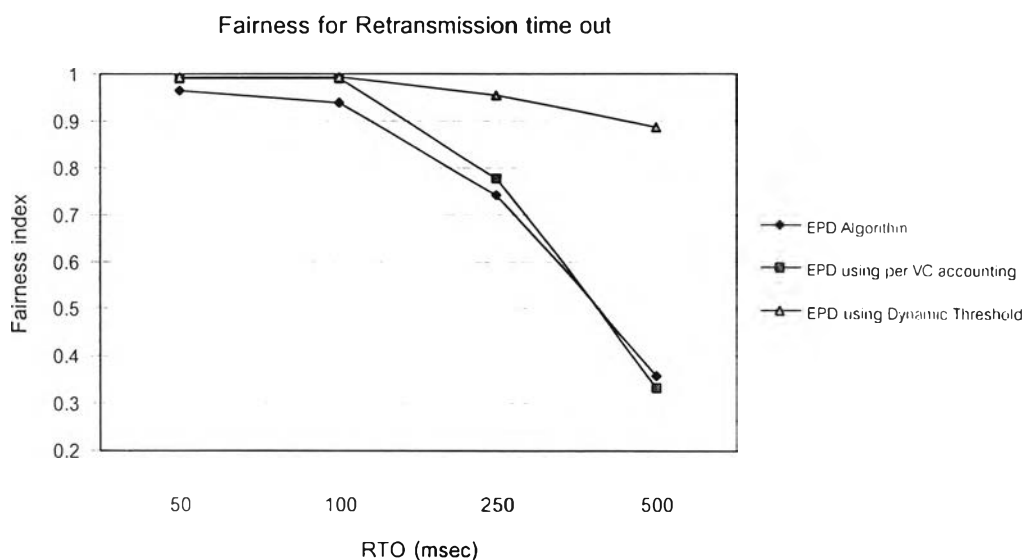
จากผลการจำลองการเปลี่ยนค่า Retransmission Time Out พบว่าเมื่อเพิ่มค่า Retransmission Time Out จะทำให้ค่าความเท่าเทียมกันและประสิทธิภาพลดลง เมื่อพิจารณาในกรณีการลดลงของประสิทธิภาพ ค่า Retransmission Time Out เพิ่มขึ้นทำให้ประสิทธิภาพลดลงเนื่องมาจากค่า Retransmission Time Out มากเมื่อความยาวคิวในบัฟเฟอร์มาถึงจุดเริ่มเปลี่ยน EPD หรือจุดเริ่มเปลี่ยนด้านสูง (อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต) สวิตช์จะทำการทิ้งเซลล์ออกไปแต่การทิ้งเซลล์ออกไปนี้สวิตช์จะทิ้งเซลล์ของทุกาแหล่งกำเนิด ดังแสดงในรูปที่ 5.48 (ค), 5.52 (ค) และ 5.56 (ค) จากนั้นเมื่อปลายทางรู้ว่ามีส่วนแพ็กเก็ตถูกทิ้งออกไป ปลายทางจะส่ง ACK ไปบอกให้ต้นทางรู้ว่ามีส่วนแพ็กเก็ตสูญหายและให้ต้นทางส่งแพ็กเก็ตที่สูญหายไปมาใหม่ แต่เวลาที่ใช้ในการรอส่งแพ็กเก็ตใหม่นั้นนานจึงทำให้ความยาวคิวลดลงจนเป็นศูนย์คือ ไม่มีเซลล์ค้างอยู่ในบัฟเฟอร์ของสวิตช์และแหล่งกำเนิดหรือต้นทางยังไม่ส่งแพ็กเก็ตมาใหม่จึงทำให้ link ในโครงข่ายว่างจึงเป็นการใช้แบนด์วิดท์ในโครงข่ายที่ไม่เหมาะสม ฉะนั้นเป็นสาเหตุให้ประสิทธิภาพของ TCP ลดต่ำกว่าเมื่อตั้งค่า Retransmission Time Out ใวน้อย เมื่อตั้งค่า Retransmission Time Out น้อยก็จะทำให้ปลายทางรู้ว่ามีส่วนแพ็กเก็ตสูญหายได้รวดเร็วและใช้เวลาการส่งแพ็กเก็ตใหม่ได้น้อยกว่า จึงทำให้มีการใช้แบนด์วิดท์ของโครงข่ายดีขึ้น

เมื่อพิจารณารูปที่ 5.45 (ค) ถึง 5.56 (ค) พบว่าขนาด Congestion window ของ Retransmission Time Out เท่ากับ 50 และ 100 ms จะมีการเพิ่มขึ้นและลดลงไม่พร้อมกันเนื่องมาจากสวิตช์มีการทิ้งเซลล์ของแต่ละแหล่งกำเนิดไม่พร้อมกัน สาเหตุที่สวิตช์ทิ้งเซลล์ของแต่ละแหล่งกำเนิดไม่พร้อมกันเพราะค่า Retransmission Time Out น้อยทำให้ปลายทางตรวจจับแพ็กเก็ตสูญหายได้รวดเร็วจึงทำให้ส่งแพ็กเก็ตใหม่ได้รวดเร็วด้วย ถ้า Retransmission Time Out เท่ากับ 250 และ 500 ms ขนาด Congestion window จะมีการเพิ่มขึ้นและลดลงเกือบจะพร้อมกัน (เรียกขนาดของ Congestion window ที่เพิ่มขึ้นหรือลดลงพร้อมกันว่า “TCP synchronization” [6]) เนื่องจากเมื่อสวิตช์ทำการทิ้งเซลล์ออกไปและกว่าต้นทางจะรู้ว่ามีส่วนแพ็กเก็ตสูญหาย ต้นทางจะรอจนถึงเวลา Time Out จึงจะเริ่มส่งแพ็กเก็ตที่สูญหายไปมาใหม่ แต่เวลาที่รอจะทำการส่งแพ็กเก็ตใหม่นั้นนานจึงดูเหมือนว่าขนาด Congestion window เพิ่มขึ้นหรือลดลงพร้อมกัน

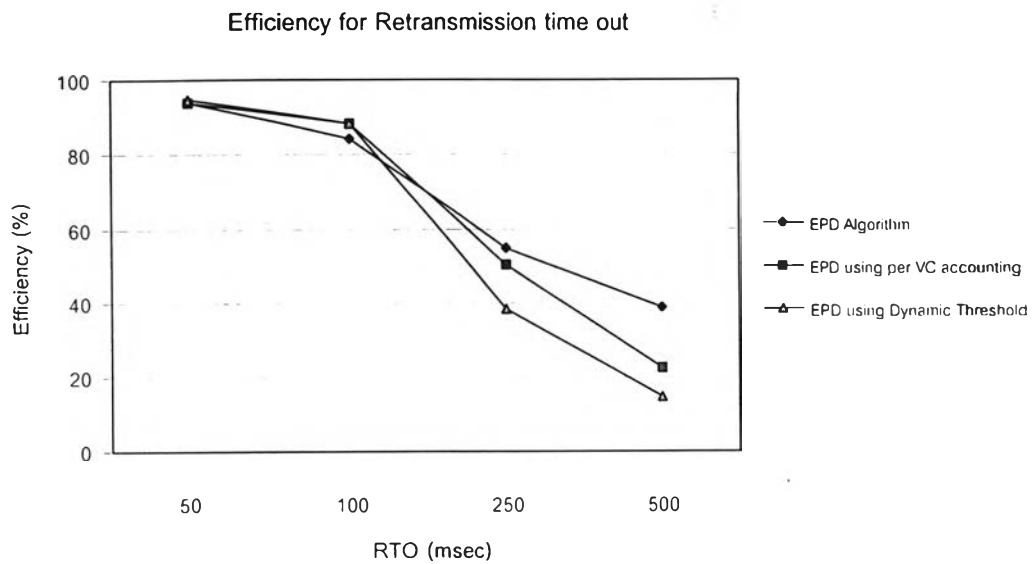
เมื่อพิจารณาในกรณีค่าความเท่าเทียมกันในการส่งข้อมูล ค่า Retransmission Time Out มากขึ้นจะมีความเท่าเทียมกันลดลงเนื่องมาจากเมื่อค่า Retransmission Time Out น้อยต้นทางจะใช้เวลารอที่จะส่งแพ็กเก็ตที่สูญหายใหม่น้อยกว่าทำให้ขนาด Congestion window มีการเพิ่มขึ้นและลดลงไม่พร้อมกัน ดังแสดงในรูปที่ 5.45 (ค), 5.49 (ค) และ 5.53 (ค) จึงทำให้ต้นทางของแต่ละ VC ทำการส่งแพ็กเก็ตใหม่ได้รวดเร็ว ดังนั้นจึงเป็นเหตุให้มีความเท่าเทียมในการส่งข้อมูล แต่ถ้าค่า Retransmission Time Out มากเมื่อสวิตช์ทำการทิ้งเซลล์ออกไป ของแหล่งกำเนิดที่มีเซลล์ถูกทิ้งก็จะทำการรอรับ ACK จากปลายทางหรือรอให้หมดเวลา time out ซึ่งเป็นเวลานานจึงจะส่งแพ็กเก็ตใหม่ แต่แหล่งกำเนิดที่ไม่มีเซลล์ถูกทิ้งออกไปจากสวิตช์ก็จะทำการส่งแพ็กเก็ตได้ต่อไปเรื่อยๆ ด้วยเหตุนี้จึงทำให้เมื่อค่า Retransmission Time Out มากทำให้ค่าความเท่าเทียมกันลดลง

จากผลการจำลองการเปลี่ยนค่า Retransmission Time Out นำมาวาดกราฟในรูปที่ 5.57 และ 5.58 ซึ่งในรูปที่ 5.57 แสดงผลการเปรียบเทียบค่าความเท่าเทียมกัน (Fairness) ของอัลกอริทึม EPD ทั้ง 3 เมื่อมีการเปลี่ยนแปลงค่า Retransmission Time Out ส่วนรูปที่ 5.58 แสดงการเปรียบเทียบประสิทธิภาพ TCP ของอัลกอริทึม EPD ทั้ง 3 เมื่อมีการเปลี่ยนแปลงค่า Retransmission Time Out

ในรูปที่ 5.58 พบว่าเมื่อค่า Retransmission Time Out เท่ากับ 250 และ 500 ms ของ และอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต มีค่าประสิทธิภาพ TCP ต่ำกว่าอัลกอริทึม EPD เพราะอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตใช้วิธีหารระดับ fair share และทิ้งเซสท์ออกไปเมื่อความยาวคิวแต่ละ VC มากกว่าระดับ fair share จึงทำให้มีการทิ้งเซสท์เป็นจำนวนมากเมื่อเทียบกับอัลกอริทึม EPD เพื่อทำให้ค่าความเท่าเทียมกันมากขึ้น ดังแสดงในรูปที่ 5.58 ด้วยเหตุนี้จึงเป็นผลให้ประสิทธิภาพของ TCP ต่ำกว่าอัลกอริทึม EPD



รูปที่ 5.57 การเปรียบเทียบความเท่าเทียมของการส่งข้อมูลของแต่ละอัลกอริทึม



รูปที่ 5.58 การเปรียบเทียบประสิทธิภาพของโครงข่ายของแต่ละอัลกอริทึม

5.5.4 ผลการจำลองการใช้กลไก Fast retransmission และ recovery ของ TCP

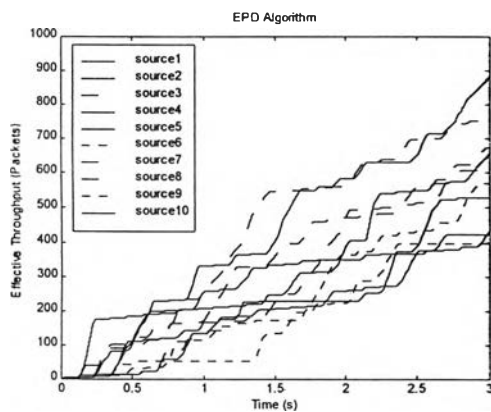
จากผลการจำลองการทำงานของอัลกอริทึม EPD, อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting และ อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต บนแบบจำลอง Peer to Peer, Parking Lot และ Chain พบว่าในบางครั้งประสิทธิภาพของ TCP ยังไม่มากเพียงพอโดยเฉพาะเมื่อเกิดแพ็กเก็ตสูญหาย ในกลไกพื้นฐานของ TCP เมื่อเกิดการทิ้งแพ็กเก็ตออกไปที่สวิตช์ TCP จะต้องรอให้หมดเวลา (time out) ก่อนปลายทางจึงจะรู้ว่ามีการสูญหาย ทำให้เส้นทางรู้ว่ามีการสูญหายช้ามาก จึงได้มีการเสนอกฎเพื่อปรับปรุงให้ปลายทางสามารถตรวจจับแพ็กเก็ตที่สูญหายและค้นหาเส้นทางส่งแพ็กเก็ตที่สูญหายใหม่ได้เร็วขึ้น เรียกกลไกนี้ว่า “Fast retransmission and recovery” ซึ่งรายละเอียดของกลไก Fast retransmission และ recovery ได้กล่าวไว้ในหัวข้อ 2.2.4

ในผลการจำลองนี้จะทำการศึกษาผลกระทบของกลไก Fast retransmission และ recovery ในโพรโตคอล TCP บนแบบจำลอง Peer to Peer และทำการศึกษาว่าเมื่อใช้กลไก Fast retransmission และ recovery จะสามารถปรับปรุงสมรรถนะของ TCP ได้หรือไม่เพียงใด โดยเงื่อนไขของแบบจำลองในการจำลองนี้จะเหมือนกับในส่วน 4.1.1 แต่จะมีกลไก Fast retransmission และ recovery ในโพรโตคอล TCP เพิ่มขึ้น

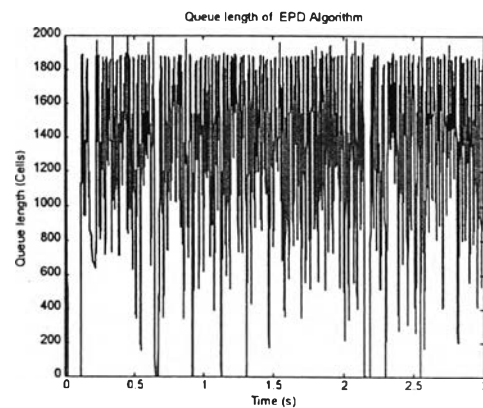
ตารางที่ 5.22 แสดงผลการจำลองการทำงานของอัลกอริทึม EPD, อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting และอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตบนแบบจำลอง Peer to Peer เมื่อใช้กลไก Fast retransmission และ recovery รูปที่ 5.59 แสดง (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวของอัลกอริทึม EPD เมื่อใช้กลไก Fast retransmission และ recovery รูปที่ 5.60 แสดง (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting เมื่อใช้กลไก Fast retransmission และ recovery และรูปที่ 5.59 แสดง (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต เมื่อใช้กลไก Fast retransmission และ recovery

ตารางที่ 5.22 ผลการจำลองของแบบจำลอง Peer to Peer เมื่อใช้กลไก Fast retransmission และ recovery

| Algorithm | Early Packet Discard (EPD) | EPD using per VG accounting | EPD using Dynamic Threshold |
|--------------------------------|----------------------------|-----------------------------|-----------------------------|
| Effective Throughput (packets) | 5941 | 5963 | 5995 |
| TCP Throughput (Mbit/s) | 128.47 | 128.15 | 129.03 |
| Fairness index | 0.9399 | 0.9762 | 0.9873 |
| Efficiency (%) | 94.90 | 94.67 | 95.38 |

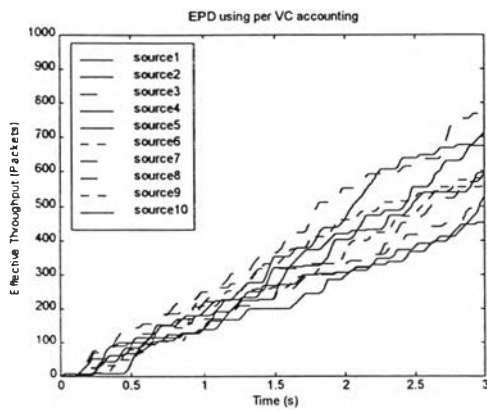


(ก)

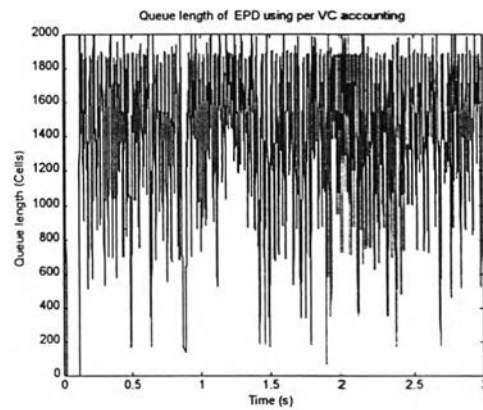


(ข)

รูปที่ 5.59 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวของอัลกอริทึม EPD เมื่อใช้กลไก Fast retransmission และ recovery

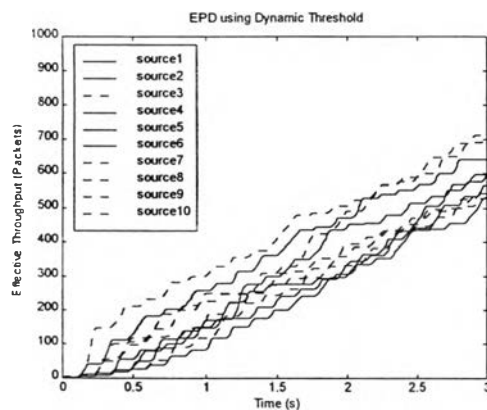


(ก)

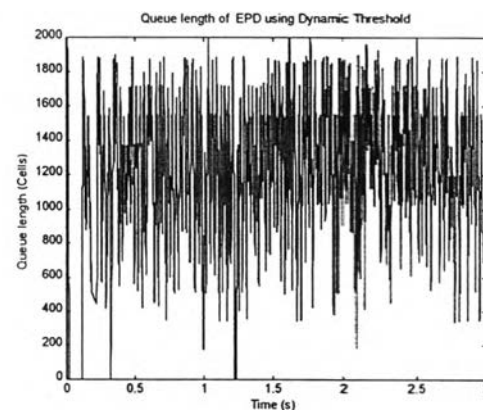


(ข)

รูปที่ 5.60 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting เมื่อใช้กลไก Fast retransmission และ recovery



(ก)



(ข)

รูปที่ 5.61 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิวของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต เมื่อใช้กลไก Fast retransmission และ recovery

การวิเคราะห์ผลการจำลองโดยใช้กลไก Fast retransmission และ recovery

ผลการจำลองการทำงานโดยใช้กลไก Fast retransmission และ recovery ในโพรโตคอล TCP จะทำให้ประสิทธิภาพของ TCP ดีขึ้นกว่าเดิม ในอัลกอริทึม EPD การใช้กลไก Fast retransmission และ recovery สามารถปรับปรุงประสิทธิภาพของ TCP แต่ค่าความเท่าเทียมกันลดลงกว่าไม่มีการใช้กลไก Fast retransmission และ recovery ในโพรโตคอล TCP เมื่อมีการใช้กลไก Fast retransmission จะทำให้เส้นทางสามารถตรวจจับแพ็กเก็ตที่สูญหายไปได้เร็วกว่า ซึ่งโดยปกติแล้วโพรโตคอล TCP จะใช้เพียงกลไก slow start และ congestion avoidance เท่านั้น ทำให้เส้นทางไม่สามารถตรวจจับและส่งแพ็กเก็ตที่สูญหายได้อย่างรวดเร็ว ถ้าเส้นทางได้รับ duplicate ACK 3 ครั้ง เส้นทางก็จะรู้ว่ามิแพ็กเก็ตสูญหายขึ้นในโครงข่ายและจะทำการส่งแพ็กเก็ตที่สูญหายใหม่ในทันทีโดยไม่ต้องรอให้หมดเวลา (time out) เส้นทางจะเริ่มส่งแพ็กเก็ตใหม่ในระยะ congestion avoidance แทนที่ระยะ slow start และไม่ได้เริ่ม congestion window เป็น 1 segment แต่จะเริ่มที่ congestion window ลดลงครึ่งหนึ่ง. ด้วยเหตุนี้จึงทำให้ผลการจำลองที่มีการใช้กลไก Fast retransmission และ recovery มีประสิทธิภาพที่ดีขึ้นกว่าเดิม โดยอัลกอริทึม EPD, อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting และอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต สามารถปรับปรุงประสิทธิภาพได้เพิ่มขึ้น 7.76%, 9.39% และ 2.75% ตามลำดับ

ผลการจำลองของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ประสิทธิภาพจะต่ำกว่าอัลกอริทึมอื่นๆ เพราะอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting มีความยาวคิวที่ล้นเกินขนาดบัฟเฟอร์ทำให้เกิดปัญหาแพ็กเก็ตที่ไม่สมบูรณ์ถูกส่งเข้าไปในโครงข่าย ดังนั้นแบนด์วิดท์ของโครงข่ายจึงมีการใช้งานอย่างไม่เต็มประสิทธิภาพ แต่อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ก็สามารถปรับปรุงความเท่าเทียมกันในการส่งข้อมูลของแต่ละแหล่งกำเนิดได้ดีกว่าอัลกอริทึม EPD เพราะมีการนับจำนวนเซลล์ที่อยู่ในสวิตช์ ถ้า VC ใดมีจำนวนเซลล์มากกว่าระดับ fair share สวิตช์ก็จะทำทิ้งเซลล์ออกไปทำให้แต่ละแหล่งกำเนิดมีความเท่าเทียมกันในการส่งข้อมูล

ส่วนผลการจำลองของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต เมื่อใช้กลไก Fast retransmission และ recovery แล้วสามารถปรับปรุงประสิทธิภาพของ TCP และความเท่าเทียมกันได้ดีขึ้นกว่าการไม่ใช้กลไก Fast retransmission และ recovery และอัลกอริทึม EPD แบบอื่นๆ ประสิทธิภาพที่เพิ่มขึ้นนี้เกิดเนื่องมาจากอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัตจะมีการใช้บัฟเฟอร์ของสวิตช์อย่างมีประสิทธิภาพโดยจะมีความยาวคิวที่ล้นออกจากรับฟเฟอร์น้อยกว่า (ดังแสดงในรูปที่ 5.61) และเมื่อมีการทิ้งเซลล์ออกจากสวิตช์ก็จะเลิกการทิ้งเซลล์ออกไปแต่ละ VC อย่างพอดีทำให้ความยาวคิวในบัฟเฟอร์ของสวิตช์ไม่ลดลงต่ำจนเหลือศูนย์ ซึ่งจะเป็นผลให้มีการใช้แบนด์วิดท์ต่ำไปด้วย แต่การเพิ่มขึ้นของประสิทธิภาพจะไม่มากนักเมื่อเปรียบเทียบกับอัลกอริทึม EPD อื่นๆ เนื่องจากอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนมีการทิ้งเซลล์ออกไปเป็นจำนวนมากกว่า

5.6 ผลการหาระดับจุดเริ่มเปลี่ยนและขนาดบัฟเฟอร์

จากการจำลองหาผลกระทบของจำนวนแหล่งกำเนิด เราสามารถสรุปได้ว่าจำนวนแหล่งกำเนิดมีผลกระทบกับขนาดของบัฟเฟอร์ที่ใช้ในสวิตช์และการตั้งระดับจุดเริ่มเปลี่ยน EPD หรือจุดเริ่มเปลี่ยนด้านสูง (อัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต) ดังที่ได้กล่าวมาแล้วในหัวข้อที่ 5.5.1 ซึ่งผลกระทบทั้งสองนี้จะมีผลทำให้ประสิทธิภาพของ TCP และค่าความเท่าเทียมกันมากขึ้นหรือลดลงได้ เช่น ถ้าขนาดบัฟเฟอร์ของสวิตช์น้อยกว่าความยาวคิวในขณะเริ่มต้นจะทำให้เกิดการล้นที่สวิตช์เป็นผลให้ประสิทธิภาพของ TCP ลดลงไปด้วย จากการจำลองในส่วนที่ 5.5.1 ซึ่งเป็นการศึกษาหาผลกระทบของจำนวนแหล่งกำเนิด ซึ่งพบว่ามีย่อไรบ้างที่เป็นผลข้างเคียง ฉะนั้นในส่วนนี้จะเป็นการจำลองหาขนาดบัฟเฟอร์ของแต่ละจำนวนแหล่งกำเนิดหรือจำนวน TCP connections ว่าจะมีขนาดเท่าใดจึงจะไม่ทำให้เกิดค่าความเท่าเทียมกันต่ำมาก และต่อจากนั้นจะทำการจำลองเพื่อศึกษาหาระดับจุดเริ่มเปลี่ยนเป็นเท่าใดจึงจะทำให้ไม่มีการสูญเสียแพ็กเกจขึ้น

5.6.1 ผลการจำลองขนาดบัฟเฟอร์ที่น้อยที่สุดของแต่ละ TCP connection

จากหัวข้อที่ 5.5.1 ผลการจำลองของ 100 TCP connections จะมีค่าความเท่าเทียมกันต่ำเป็น 0.01 เท่านั้นจึงได้มีการหาสาเหตุว่าที่ 100 TCP connections ทำไมจึงมีค่าความเท่าเทียมต่ำ และสามารถสรุปหาสาเหตุได้ดังนี้ เมื่อขนาดของบัฟเฟอร์เป็นไปตามสมการที่ (5.1) จะทำให้ค่าความเท่าเทียมต่ำ

$$Buffer\ size < MTU \times connections \quad (5.1)$$

ขนาดบัฟเฟอร์ที่น้อยที่สุดที่ค่าความเท่าเทียมต่ำเราเรียกว่า “จุดวิกฤต” จุดวิกฤตนี้เป็นจุดที่มีค่าความเท่าเทียมกันในการส่งข้อมูลต่ำมาก เราจึงพยายามหลีกเลี่ยงที่จะให้โครงข่ายอยู่ในสภาวะนี้ โดยในแต่ละจำนวน TCP connections จะต้องมีจุดนี้เกิดขึ้นทุกๆจำนวน TCP connections เช่น 5 TCP connections ก็จะมีขนาดบัฟเฟอร์ขนาดหนึ่งที่ทำให้เกิดจุดวิกฤต ส่วน 10 TCP connections ก็จะมีขนาดบัฟเฟอร์ขนาดหนึ่งที่ทำให้เกิดจุดวิกฤตเหมือนกัน แต่ขนาดบัฟเฟอร์ที่ทำให้เกิดจุดวิกฤตของ 5 TCP connections และ 10 TCP connections จะไม่เท่ากัน ฉะนั้นในส่วนนี้จึงเป็นการจำลองหาขนาดบัฟเฟอร์ที่น้อยที่สุดของแต่ละ TCP connections ที่ทำให้ความเท่าเทียมกันไม่ถึงจุดวิกฤต ซึ่งจะเป็นข้อมูลเบื้องต้นในการเลือกขนาดบัฟเฟอร์ของสวิตช์เพื่อหลีกเลี่ยงไม่ให้โครงข่ายอยู่ในสภาวะนี้ โดยในตอนแรกจะทำการเปรียบเทียบขนาดบัฟเฟอร์ที่น้อยที่สุดที่ไม่ทำให้เกิดค่าความเท่าเทียมอยู่ที่จุดวิกฤตจากการคำนวณตามสมการที่ (5.1) และจากการจำลอง ส่วนต่อไปจะเป็นผลการจำลองขนาดบัฟเฟอร์ของแต่ละจำนวนแหล่งกำเนิดของทั้ง 3 อัลกอริทึม

ในตารางที่ 5.23 แสดงผลการเปรียบเทียบขนาดของบัฟเฟอร์น้อยที่สุดที่ไม่ทำให้ค่าความเท่าเทียมกันอยู่ที่จุดวิกฤตจากการคำนวณและการจำลอง และตารางที่ 5.24 ถึง 5.26 แสดงผลการจำลองหาขนาดบัฟเฟอร์ที่จุดวิกฤตของแต่ละ TCP connections ของอัลกอริทึม EPD, อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting, และอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต ตามลำดับ ส่วนรูปที่ 5.62 แสดงผลการเปรียบเทียบประสิทธิภาพ TCP ของขนาดบัฟเฟอร์ที่จุดวิกฤตและที่พ้นจุดวิกฤตของแต่ละอัลกอริทึม รูปที่ 5.63 แสดงผลการเปรียบเทียบค่าความเท่าเทียมกันของขนาดบัฟเฟอร์ที่จุดวิกฤตและที่พ้นจุดวิกฤต รูปที่ 5.64 และ 5.65 แสดงผลการจำลองของขนาดบัฟเฟอร์ที่จุดวิกฤตและที่พ้นจุดวิกฤต ตามลำดับ ในรูปที่ 5.64 และ 5.65 เป็นผลการจำลองขนาดบัฟเฟอร์ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต ในส่วนของอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting ก็จะได้รูปกราฟที่เหมือนกัน

ตารางที่ 5.23 ผลการเปรียบเทียบขนาดของบัฟเฟอร์น้อยที่สุดที่ทำให้ความเท่าเทียมกัน ไม่วิกฤต โดยการคำนวณและการจำลอง

| Number of TCP connection | Minimum buffer size from calculate (cells) | Minimum buffer size from simulation (cells) |
|--------------------------|---|--|
| 5 | 855 | 665 |
| 10 | 1710 | 1488 |
| 15 | 2565 | 2306 |
| 25 | 4275 | 3990 |
| 50 | 8550 | 8089 |
| 100 | 17100 | 16305 |

ตารางที่ 5.24 ผลการจำลองขนาดบัฟเฟอร์ที่จุดวิกฤตของแต่ละ connection ของอัลกอริทึม EPD

| Number of TCP connection | 5 | | 10 | | 15 | | 25 | | 50 | | 100 | |
|---------------------------------|--------|-------|---------|-------|--------|--------|---------|--------|---------|---------|---------|--------|
| Buffer size (cells) | 844 | 665 | 1487 | 1488 | 2305 | 2306 | 3989 | 3990 | 8088 | 8089 | 16304 | 16305 |
| Effective throughput (packets) | 6153 | 704 | 6148 | 1482 | 6143 | 3264 | 6133 | 4675 | 6109 | 6348 | 6062 | 6298 |
| Total TCP throughput (Mbit/sec) | 133.55 | 15.09 | 133.446 | 31.84 | 133.34 | 69.98 | 133.131 | 100.07 | 132.615 | 133.391 | 131.578 | 132.32 |
| Total loss (cells) | 16 | 10275 | 50 | 9967 | 84 | 7465 | 106 | 5335 | 286 | 285 | 620 | 619 |
| Fairness | 0.2 | 0.4 | 0.1 | 0.2 | 0.0667 | 0.1327 | 0.04 | 0.0798 | 0.02 | 0.04 | 0.01 | 0.02 |
| Efficiency (%) | 98.66 | 11.15 | 98.58 | 23.52 | 98.50 | 51.69 | 98.35 | 73.92 | 97.96 | 98.54 | 97.20 | 97.75 |

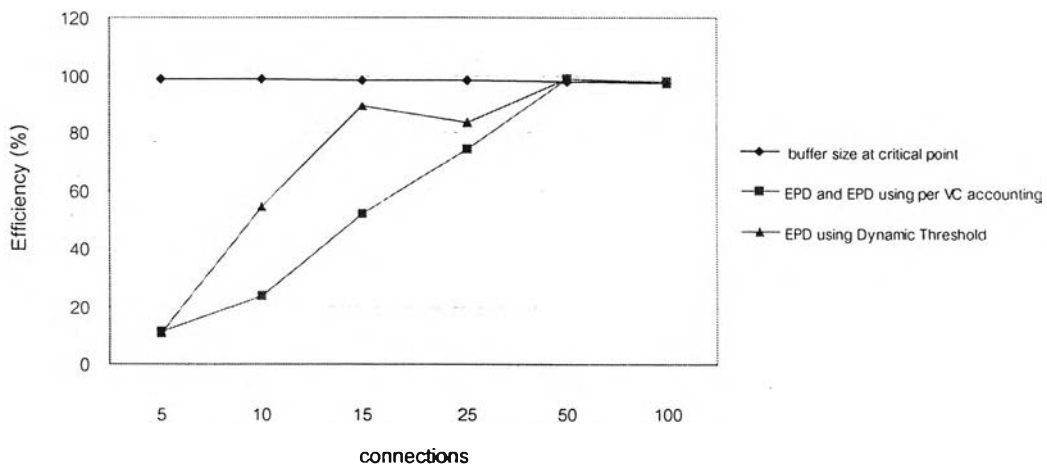
ตารางที่ 5.25 ผลการจำลองขนาดบัฟเฟอร์ที่จุดวิกฤตของแต่ละ connection ของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting

| Number of TCP connection | 5 | | 10 | | 15 | | 25 | | 50 | | 100 | |
|---------------------------------|--------|--------|---------|-------|--------|--------|--------|---------|---------|---------|---------|--------|
| Buffer size (cells) | 664 | 665 | 1487 | 1488 | 2305 | 2306 | 3989 | 3990 | 8088 | 8089 | 16304 | 16305 |
| Effective throughput (packets) | 6153 | 704 | 6148 | 1482 | 6143 | 3264 | 6133 | 4675 | 6109 | 6348 | 6062 | 6298 |
| Total TCP throughput (Mbit/sec) | 133.55 | 15.089 | 133.446 | 31.84 | 133.34 | 69.98 | 133.13 | 100.067 | 132.615 | 133.391 | 131.578 | 132.32 |
| Total loss (cells) | 16 | 10275 | 50 | 9967 | 84 | 7465 | 106 | 5335 | 286 | 285 | 620 | 619 |
| Fairness | 0.2 | 0.4 | 0.1 | 0.2 | 0.0667 | 0.1327 | 0.04 | 0.0798 | 0.02 | 0.04 | 0.01 | 0.02 |
| Efficiency (%) | 98.66 | 11.15 | 98.58 | 23.52 | 98.50 | 51.69 | 98.35 | 73.92 | 97.96 | 98.54 | 97.20 | 97.75 |

ตารางที่ 5.26 ผลการจำลองขนาดบัฟเฟอร์ที่จุดวิกฤตของแต่ละ connection ของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต

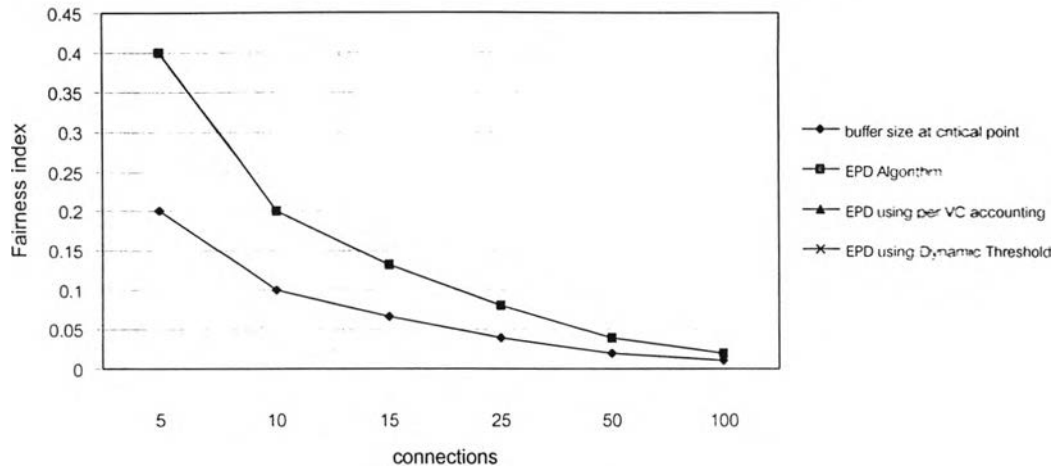
| Number of TCP connection | 5 | | 10 | | 15 | | 25 | | 50 | | 100 | |
|---------------------------------|--------|-------|---------|-------|--------|--------|---------|---------|---------|---------|---------|--------|
| Buffer size (cells) | 644 | 665 | 1487 | 1488 | 2305 | 2306 | 3989 | 3990 | 8088 | 8089 | 16304 | 16305 |
| Effective throughput (packets) | 6153 | 699 | 6148 | 3410 | 6143 | 5603 | 6133 | 5237 | 6109 | 6348 | 6062 | 6298 |
| Total TCP throughput (Mbit/sec) | 133.55 | 15.02 | 133.446 | 73.31 | 133.34 | 120.76 | 133.131 | 113.075 | 132.615 | 133.391 | 131.578 | 132.32 |
| Total loss (cells) | 16 | 10959 | 50 | 12019 | 84 | 8747 | 106 | 5113 | 286 | 285 | 620 | 619 |
| Fairness | 0.2 | 0.4 | 0.1 | 0.2 | 0.0667 | 0.1333 | 0.04 | 0.08 | 0.02 | 0.04 | 0.01 | 0.02 |
| Efficiency (%) | 98.66 | 11.10 | 98.58 | 54.15 | 98.50 | 89.21 | 98.35 | 83.53 | 97.96 | 98.54 | 97.20 | 97.20 |

Efficiency for buffer size and number of TCP connection of each algorithm

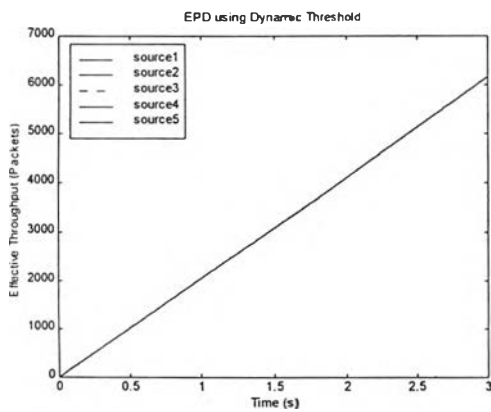


รูปที่ 5.62 ผลการเปรียบเทียบประสิทธิภาพของขนาดบัฟเฟอร์ที่จุดวิกฤตและที่พ้นจุดวิกฤต

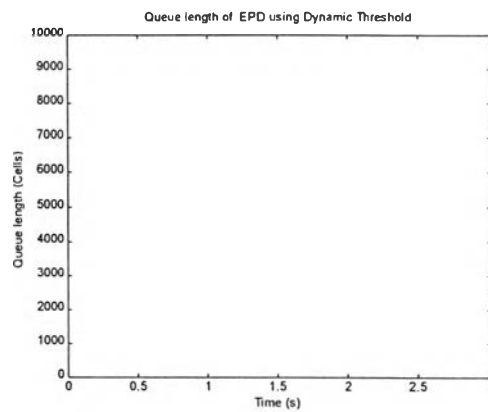
Fairness for buffer size and number of TCP connection of each algorithm



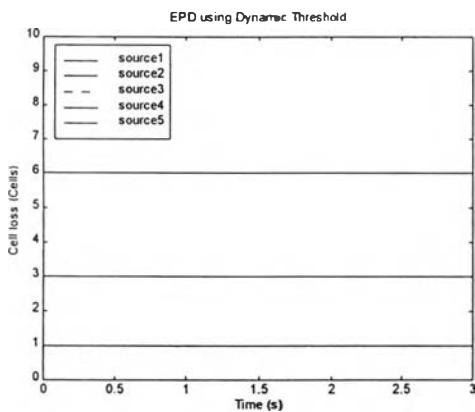
รูปที่ 5.63 ผลการเปรียบเทียบความเท่าเทียมกันของขนาดบัฟเฟอร์ที่จุดวิกฤตและพื้นที่วิกฤต



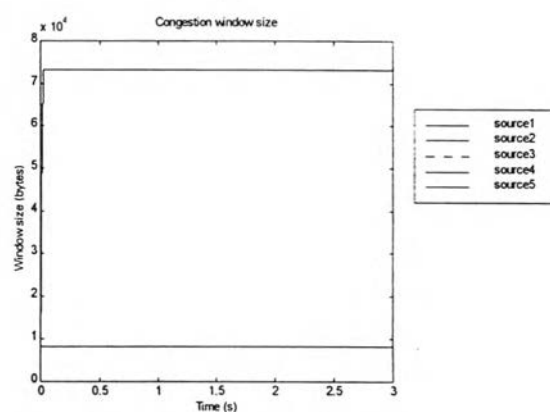
(ก)



(ข)

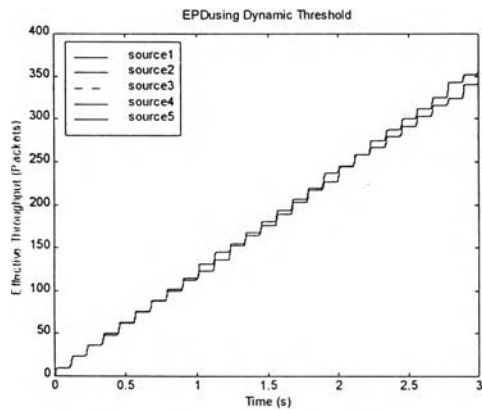


(ค)

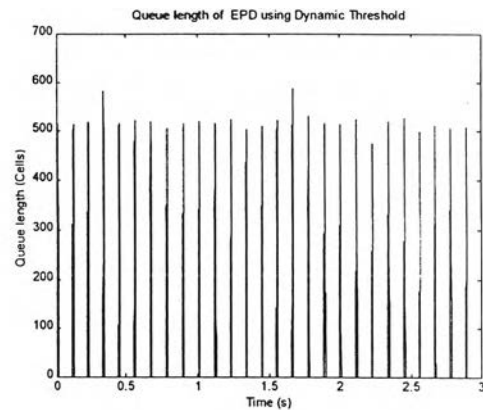


(ง)

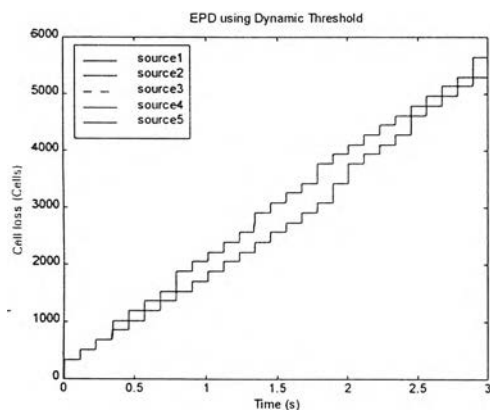
รูปที่ 5.64 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิว (ค) จำนวนเซลล์ที่สูญเสีย (ง) ขนาด Congestion window ที่ขนาดบัฟเฟอร์อยู่บนจุดวิกฤต



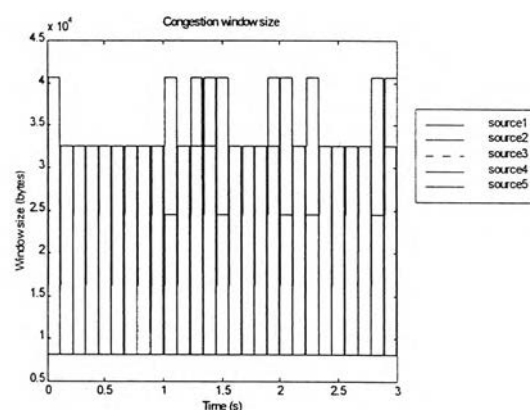
(ก)



(ข)



(ค)



(ง)

รูปที่ 5.65 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิว (ค) จำนวนเซลล์ที่สูญเสีย
(ง) ขนาด Congestion window ที่ขนาดบัฟเฟอร์พื้นฐานจุดวิกฤต

การวิเคราะห์หาขนาดบัฟเฟอร์ของแต่ละ TCP connections

ถ้าขนาดบัฟเฟอร์ต่ำกว่าหรือเท่ากับจุดวิกฤต (ในที่นี้คือ 644, 1487, 2305, 3889, 8088 และ 16304) จะมีเพียง VC เดียวที่สามารถส่งข้อมูลได้ ส่วน VC อื่นๆจะส่งข้อมูลได้ไม่ถึงหนึ่งแพ็กเกจ เพราะความยาวคิวขณะเริ่มต้นมากเกินกว่าขนาดบัฟเฟอร์ของสวิตช์ จึงทำให้มีการทิ้งเซกซ์ออกไป VC ใดที่ถูกทิ้งเซกซ์ออกไป โพรโทคอล TCP ก็จะดำเนินการรอ retransmit แพ็กเกจเดิมที่ถูกทิ้งออกไป ส่วน VC ที่ไม่ถูกทิ้งเซกซ์ออกไป ขณะเริ่มต้นก็ยังสามารถส่งข้อมูลได้ต่อไปเรื่อยๆ ดังแสดงในรูปที่ 5.64 ดังนั้นจะเห็นว่ามียูเพียง VC เดียวที่สามารถส่งข้อมูลได้และทำให้ค่าความเท่าเทียมกันต่ำ ซึ่งค่าความเท่าเทียมกันสามารถสรุปได้ดังนี้เมื่อขนาดบัฟเฟอร์ต่ำกว่าหรือเท่ากับจุดวิกฤต

$$\text{ค่าความเท่าเทียมกัน (Fairness)} = \frac{1}{\text{จำนวน TCP connections}}$$

ถ้าขนาดบัฟเฟอร์มากกว่าจุดวิกฤต จะมี 2 VC ที่สามารถส่งข้อมูลได้ ส่วน VC ที่เหลือส่งข้อมูลได้ไม่ถึงหนึ่งแพ็กเกจ โดยจะถูกทิ้งออกไปที่สวิตช์ ดังแสดงในรูปที่ 5.65 ส่วน VC ที่ส่งข้อมูลได้ไม่ถึงหนึ่งแพ็กเกจจะไม่สามารถส่งข้อมูลได้อีก เพราะโพรโทคอล TCP จะทำการรอ retransmit แพ็กเกจเดิมใหม่ ส่วน VC ทั้งสองที่สามารถส่งข้อมูลได้ครบหนึ่งแพ็กเกจในตอนแรกก็จะสามารถส่งข้อมูลได้ต่อไปเรื่อยๆ เพราะไม่ต้องรอการ retransmit ใหม่ จากผลการจำลองขนาดบัฟเฟอร์ที่มากกว่าจุดวิกฤตจะมี VC ที่สามารถส่งข้อมูลได้เพิ่มขึ้นอีกหนึ่ง VC เมื่อเทียบกับผลการจำลองขนาดบัฟเฟอร์ที่ต่ำกว่าจุดวิกฤต เนื่องจากการเพิ่มขนาดบัฟเฟอร์เพียงหนึ่งเซกซ์ โดย VC ที่เพิ่มขึ้นนี้เป็น VC ที่มีการทิ้งเซกซ์ที่สวิตช์ในขณะเริ่มต้นเพียงแค่หนึ่งเซกซ์เท่านั้นเมื่อขนาดบัฟเฟอร์ต่ำกว่าจุดวิกฤต โดยขนาดบัฟเฟอร์ที่เพิ่มขึ้นเพียงหนึ่งเซกซ์เป็นขนาดที่ไว้สำหรับรองรับ VC ที่เพิ่มขึ้นให้สามารถส่งข้อมูลได้ครบหนึ่งแพ็กเกจในขณะเริ่มต้น จากผลการจำลองขนาดบัฟเฟอร์มากกว่าจุดวิกฤต พบว่าค่าความเท่าเทียมกันในการส่งข้อมูลเป็น 2 เท่าของผลการจำลองขนาดบัฟเฟอร์ที่ต่ำกว่าหรือเท่ากับจุดวิกฤต เพราะมีจำนวน VC ที่สามารถส่งข้อมูลได้เพิ่มขึ้นอีกหนึ่ง VC ดังนั้นค่าความเท่าเทียมกันของขนาดบัฟเฟอร์ที่มากกว่าจุดวิกฤตเพียงหนึ่งเซกซ์เท่ากับ

$$\text{ค่าความเท่าเทียมกัน (Fairness)} = \frac{2}{\text{จำนวน TCP connections}}$$

เมื่อเพิ่มขนาดบัฟเฟอร์มากกว่าจุดวิกฤตไปเรื่อยๆ ก็จะมีผลทำให้ค่าความเท่าเทียมกันมากขึ้นเรื่อยๆ เหมือนกัน ส่วนค่าวิสัยสามารถของขนาดบัฟเฟอร์มากกว่าจุดวิกฤตจะต่ำกว่าค่าวิสัยสามารถของขนาดบัฟเฟอร์ต่ำกว่าหรือเท่ากับจุดวิกฤต เพราะค่าวิสัยสามารถของขนาดบัฟเฟอร์มากกว่าจุดวิกฤตต้องแบ่งระหว่างสอง VC ที่สามารถส่งข้อมูลได้ ส่วน VC อื่นๆที่ไม่สามารถส่งข้อมูลได้ก็จะมีค่าวิสัยสามารถต่ำมาก ค่าวิสัย

สามารถของขนาดบัฟเฟอร์ที่ต่ำกว่าหรือเท่ากับจุดวิกฤตมีค่าสูงเพราะในโครงข่ายมี VC เดียวที่ส่งข้อมูลได้ ดังนั้นค่าวิสัยสามารถของ VC เดียวที่ส่งข้อมูลได้จึงเกือบจะเท่ากับแบนด์วิธของโครงข่าย แต่ที่วิสัยสามารถของ VC ที่ส่งข้อมูลได้ไม่เท่ากับแบนด์วิธของโครงข่ายเนื่องจากโครงข่ายมีการทิ้งเซลล์ออกไปที่สวิตช์ในขณะเริ่มต้น

5.6.2 ผลการจำลองหาระดับจุดเริ่มเปลี่ยนโดยไม่ทำให้เกิดเซลล์สูญเสีย

จากในหัวข้อที่ 5.5.1 เป็นการศึกษาผลกระทบของจำนวนแหล่งกำเนิด ซึ่งทำให้รู้ว่าระดับจุดเริ่มเปลี่ยนของอัลกอริทึมที่สวิตช์เป็นผลกระทบหนึ่งต่อจำนวนแหล่งกำเนิด ถ้าต้องการโครงข่ายมีประสิทธิภาพดีควรจะมีการตั้งระดับจุดเริ่มเปลี่ยนที่เหมาะสม Shiv Kalyanaraman [16] ได้ศึกษาถึงขนาดบัฟเฟอร์ที่ต้องการสำหรับการบริการแบบ ABR และ UBR ในโครงข่าย ATM จากการศึกษาของ Shiv Kalyanaraman พบว่าการบริการ UBR มีความต้องการขนาดบัฟเฟอร์เท่ากับผลรวมของขนาด window ที่มากที่สุดของทุกๆ TCP connections เพื่อหลีกเลี่ยงเกิดการสูญเสียเซลล์ขึ้น ซึ่งผลการศึกษาของ Shiv Kalyanaraman สอดคล้องกับผลการจำลองและการวิเคราะห์ในหัวข้อที่ 5.5.1 โดยขนาดบัฟเฟอร์ที่ต้องการของการบริการ UBR สอดคล้องกับระดับจุดเริ่มเปลี่ยนของอัลกอริทึม EPD ผลการจำลองหาผลกระทบจำนวนแหล่งกำเนิดหรือ TCP connections สามารถสรุปเกี่ยวกับระดับจุดเริ่มเปลี่ยนที่ทำให้ไม่เกิดเซลล์สูญเสียได้ว่า

$$Threshold \geq Maximum Window size \times connections \quad (5.2)$$

ในส่วนนี้จะทำการหาระดับจุดเริ่มเปลี่ยนที่ทำให้ประสิทธิภาพสูงสุดหรือมีเซลล์สูญเสียน้อยที่สุดและมีความเท่าเทียมกันสูงสุดของแต่ละอัลกอริทึม โดยในตอนแรกจะทำการเปรียบเทียบระดับจุดเริ่มเปลี่ยนที่ได้จากการคำนวณในสมการที่ (5.2) กับผลการจำลองทั้ง 3 อัลกอริทึม ซึ่งผลการเปรียบเทียบนี้แสดงในตารางที่ 5.27 จากนั้นจึงทำการจำลองระดับจุดเริ่มเปลี่ยนของแต่ละอัลกอริทึมว่ามีค่าประสิทธิภาพและค่าความเท่าเทียมกันเป็นเท่าใด ดังแสดงในตารางที่ 5.28 ถึง 5.30 โดยตารางที่ 5.28 แสดงผลการจำลองขนาดจุดเริ่มเปลี่ยนน้อยที่สุดที่ไม่มีแพ็กเกจสูญเสียและขนาดจุดเริ่มเปลี่ยนมากที่สุดที่เกิดแพ็กเกจสูญเสียของอัลกอริทึม EPD ตารางที่ 5.29 แสดงผลการจำลองขนาดจุดเริ่มเปลี่ยนน้อยที่สุดที่ไม่มีแพ็กเกจสูญเสียและขนาดจุดเริ่มเปลี่ยนมากที่สุดที่เกิดแพ็กเกจสูญเสียของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting และตารางที่ 5.30 แสดงผลการจำลองขนาดจุดเริ่มเปลี่ยนน้อยที่สุดที่ไม่มีแพ็กเกจสูญเสียและขนาดจุดเริ่มเปลี่ยนมากที่สุดที่เกิดแพ็กเกจสูญเสียของอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต ส่วนรูปที่ 5.66 แสดงผลการจำลอง (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิว (ค) เซลล์ที่สูญเสีย (ง) Congestion Window sizes ของแต่ละแหล่งกำเนิด โดยตั้งจุดเริ่มเปลี่ยนที่ทำให้ไม่เกิดเซลล์สูญเสียและทุกแหล่งกำเนิดมีความเท่าเทียมกันในการส่งข้อมูล รูปที่ 5.67 แสดงผลการจำลอง (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิว (ค) เซลล์ที่สูญเสีย (ง) Congestion Window sizes ของแต่ละแหล่งกำเนิด โดยตั้งจุดเริ่มเปลี่ยนมากที่สุดที่ทำให้เกิดเซลล์สูญเสีย ซึ่งรูปที่ 5.66 และ 5.67 เป็นผลการจำลองของอัลกอริทึม EPD

ตารางที่ 5.27 ผลการเปรียบเทียบระดับจุดเริ่มเปลี่ยนที่ได้จากการคำนวณและจากการจำลองของแต่ละอัลกอริทึม

| Number of TCP connection | Minimum threshold from calculation (cells) | Minimum threshold from simulation (cells) | | |
|--------------------------|--|---|-----------------------------|-----------------------------|
| | | Early Packet Discard (EPD) | EPD using per VC accounting | EPD using Dynamic Threshold |
| 5 | 6911 | 6335 | 5984 | 6335 |
| 10 | 13822 | 13180 | 11970 | 13180 |
| 15 | 20734 | 20017 | 17955 | 20017 |
| 25 | 34557 | 33700 | 29589 | 33700 |
| 50 | 69114 | 67898 | 59766 | 67898 |

ตารางที่ 5.28 ผลการจำลองขนาดจุดเริ่มเปลี่ยนมากที่สุดที่ไม่มีแพ็คเกจสูญเสียของอัลกอริทึม EPD

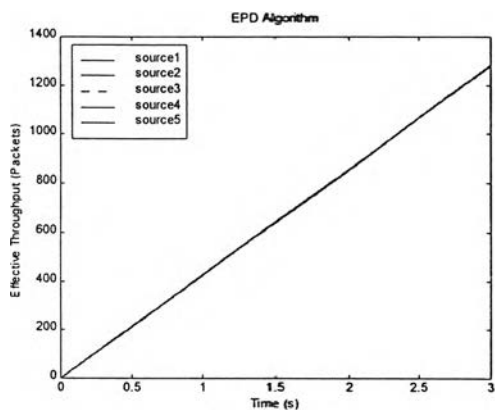
| Number of TCP connection | 5 | | 10 | | 15 | | 25 | | 50 | |
|---------------------------------|--------|--------|--------|--------|--------|--------|--------|-------|---------|---------|
| Threshold (cells) | 6334 | 6335 | 13179 | 13180 | 20016 | 20017 | 33699 | 33700 | 67897 | 67898 |
| Effective throughput (packets) | 6391 | 6392 | 6385 | 6386 | 6381 | 6379 | 6366 | 6367 | 6344 | 6345 |
| Total TCP throughput (Mbit/sec) | 134.39 | 134.39 | 134.39 | 134.39 | 134.39 | 134.42 | 134.4 | 134.4 | 134.435 | 134.441 |
| Total loss (cells) | 171 | 0 | 171 | 0 | 171 | 0 | 171 | 0 | 171 | 0 |
| Fairness | 0.9995 | 1 | 0.9995 | 1 | 0.9994 | 1 | 0.9994 | 1 | 0.998 | 0.9995 |
| Efficiency (%) | 99.28 | 99.28 | 99.27 | 99.27 | 99.28 | 99.30 | 99.30 | 99.30 | 99.30 | 99.31 |

ตารางที่ 5.29 ผลการจำลองขนาดจุดเริ่มเปลี่ยนมากที่สุดที่ไม่มีแพ็คเกจสูญเสียของอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting

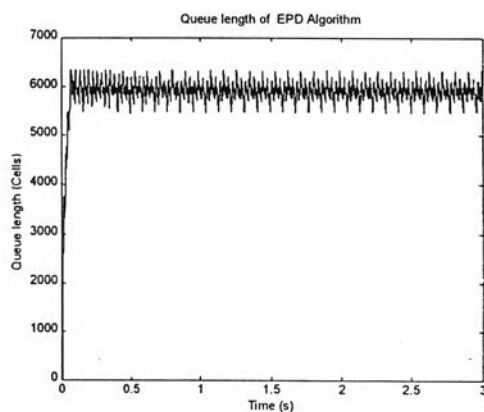
| Number of TCP connection | 5 | | 10 | | 15 | | 25 | | 50 | |
|---------------------------------|--------|--------|--------|--------|---------|--------|--------|--------|--------|---------|
| Threshold (cells) | 5983 | 5984 | 11969 | 11970 | 17954 | 17955 | 29588 | 29589 | 59766 | 59767 |
| Effective throughput (packets) | 6391 | 6392 | 6385 | 6386 | 6380 | 6379 | 6345 | 6367 | 6345 | 6345 |
| Total TCP throughput (Mbit/sec) | 134.39 | 134.39 | 134.39 | 134.39 | 134.405 | 134.42 | 134.41 | 134.41 | 134.42 | 134.441 |
| Total loss (cells) | 100 | 0 | 171 | 0 | 171 | 0 | 4104 | 0 | 171 | 0 |
| Fairness | 0.9995 | 1 | 0.9995 | 1 | 0.9995 | 1 | 0.9955 | 1 | 0.9987 | 0.9995 |
| Efficiency (%) | 99.28 | 99.28 | 99.27 | 99.27 | 99.29 | 99.30 | 99.29 | 99.29 | 99.30 | 99.31 |

ตารางที่ 5.30 ผลการจำลองขนาดจุดเริ่มเปลี่ยนมากที่สุดที่ไม่มีแพ็คเกจสูญเสียของอัลกอริทึม EPD ที่ใช้เทคนิค จุดเริ่มเปลี่ยนพลวัต

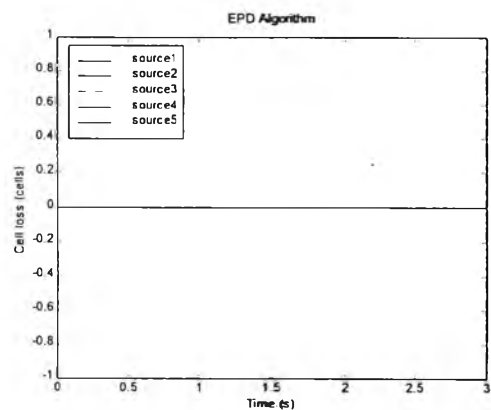
| Number of TCP connection | 5 | | 10 | | 15 | | 25 | | 50 | |
|---------------------------------|--------|--------|--------|--------|--------|--------|--------|-------|--------|---------|
| Threshold (cells) | 6334 | 6335 | 13179 | 13180 | 20016 | 20017 | 33699 | 33700 | 67897 | 67898 |
| Effective throughput (packets) | 6391 | 6335 | 6385 | 6386 | 6381 | 6379 | 6366 | 6367 | 6344 | 6345 |
| Total TCP throughput (Mbit/sec) | 134.39 | 134.39 | 134.39 | 134.39 | 134.39 | 134.42 | 134.4 | 134.4 | 134.43 | 134.441 |
| Total loss (cells) | 171 | 0 | 171 | 0 | 171 | 0 | 171 | 0 | 171 | 0 |
| Fairness | 0.9995 | 1 | 0.9995 | 1 | 0.9994 | 1 | 0.9994 | 1 | 0.998 | 0.9995 |
| Efficiency (%) | 99.28 | 99.28 | 99.27 | 99.27 | 99.28 | 99.30 | 99.30 | 99.30 | 99.30 | 99.31 |



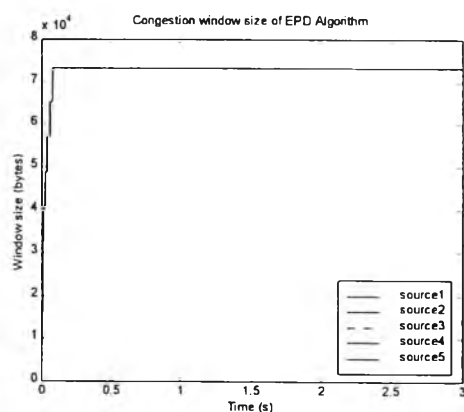
(ก)



(ข)

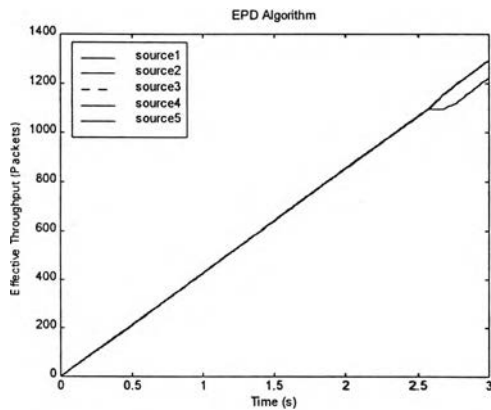


(ค)

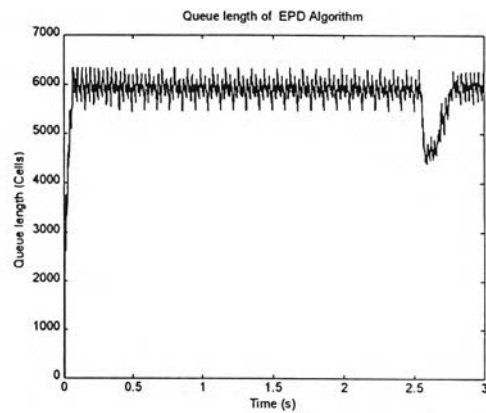


(ง)

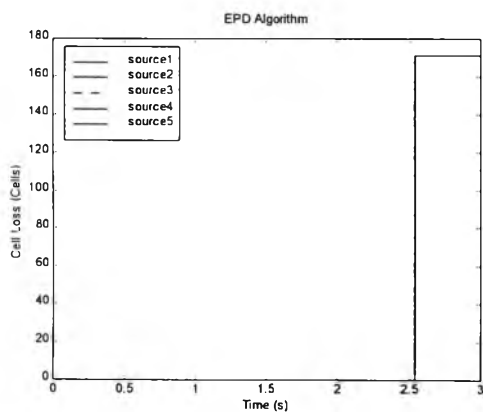
รูปที่ 5.66 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิว (ค) เซลล์ที่สูญเสีย (ง) Congestion Window sizes ของแต่ละแหล่งกำเนิด โดยตั้งจุดเริ่มเปลี่ยนที่ทำให้ไม่เกิดเซลล์สูญเสีย



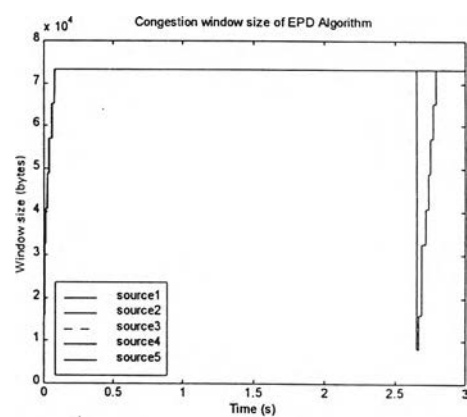
(ก)



(ข)



(ค)



(ง)

รูปที่ 5.67 (ก) Effective Throughput ของแต่ละแหล่งกำเนิด (ข) ความยาวคิว (ค) เซลล์ที่สูญเสีย (ง) Congestion Window sizes ของแต่ละแหล่งกำเนิด โดยตั้งจุดเริ่มเปลี่ยนที่ทำให้เกิดเซลล์สูญเสีย

ผลการวิเคราะห์การหาระดับจุดเริ่มเปลี่ยนที่ทำให้ประสิทธิภาพและค่าความเท่าเทียมกันสูงสุด

จากตารางที่ 5.27 แสดงผลการเปรียบเทียบระดับจุดเริ่มเปลี่ยนน้อยที่สุด โดยไม่มีเซลล์สูญเสียเกิดขึ้นและแต่ละแหล่งกำเนิดมีความเท่าเทียมกันที่ได้รับการจำลองและจากการคำนวณตามสมการที่ (5.2) จะสังเกตเห็นว่าระดับจุดเริ่มเปลี่ยนที่ได้จากการจำลองจะมีค่าน้อยกว่าระดับจุดเริ่มเปลี่ยนที่ได้จากการคำนวณเสมอ เพราะเมื่อแหล่งกำเนิดส่งข้อมูลถึงระดับขนาด window เท่ากับ 64 Kbytes แล้วข้อมูลจากแหล่งกำเนิดถูกส่งเข้าไปในสวิตช์ เมื่อถึงสวิตช์แล้วสวิตช์ก็จะมี การส่งข้อมูลออกไปในโครงข่ายตลอดจึงทำให้ความยาวคิวลดลงไม่ถึงระดับจุดเริ่มเปลี่ยนที่คำนวณได้ ดังนั้นในทางปฏิบัติการจะตั้งระดับจุดเริ่มเปลี่ยนเราควรจะเลือก ระดับจุดเริ่มเปลี่ยนที่ได้จากการคำนวณซึ่งจะรับประกันได้ว่าเป็นระดับจุดเริ่มเปลี่ยนที่ไม่ทำให้เกิดเซลล์สูญเสียขึ้นและแต่ละแหล่งกำเนิดมีความเท่าเทียมกันในการส่งข้อมูล จากผลการจำลองในตารางที่ 5.28 ถึง 5.30 ระดับจุดเริ่มเปลี่ยนของอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต จะมีค่าอยู่ที่ระดับเดียวกันเพราะ ระดับจุดเริ่มเปลี่ยนที่ใช้ในการตรวจจับการทิ้งเซลล์ของสวิตช์จะเหมือนกันคือ อัลกอริทึม EPD จะใช้จุดเริ่มเปลี่ยน EPD และอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต จะใช้จุดเริ่มเปลี่ยนด้านสูง เป็นจุดเริ่มเปลี่ยนที่ใช้ตรวจจับว่าเมื่อใดจะทำการทิ้งเซลล์ออกไป ส่วนอัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting จะมีระดับจุดเริ่มเปลี่ยนที่ต่ำกว่าอัลกอริทึม EPD และอัลกอริทึม EPD ที่ใช้เทคนิคจุดเริ่มเปลี่ยนพลวัต เพราะระดับจุดเริ่มเปลี่ยนที่ไม่ทำให้เกิดเซลล์สูญเสียจะอยู่ที่ระดับสูงหรือต่ำขึ้นอยู่กับความยาวคิว (queue length) แต่อัลกอริทึม EPD ที่ใช้เทคนิค per-VC accounting จะมีการทิ้งเซลล์เมื่อความยาวคิวของแต่ละ VC มากเกินกว่าระดับ fair share ฉะนั้นความยาวคิวจึงต่ำกว่าอัลกอริทึมทั้งสองจึงทำให้ระดับจุดเริ่มเปลี่ยนต่ำกว่าด้วย

จากตารางที่ 5.28 และ 5.30 จะสังเกตเห็นว่าถ้าระดับจุดเริ่มเปลี่ยนต่ำกว่าระดับจุดเริ่มเปลี่ยนที่ไม่เกิดเซลล์สูญเสีย (เรียกระดับจุดเริ่มเปลี่ยนที่ไม่ทำให้เกิดเซลล์สูญเสียว่า “minimum threshold”) จะมีเซลล์สูญเสียเกิดขึ้น 171 เซลล์ ซึ่งเท่ากับ 1 แพ็กเกจ ส่วนค่าวิสัยสามารถและค่าความเท่าเทียมกันไม่ต่างจากระดับ minimum threshold เท่าใดนัก แต่ในตารางที่ 5.29 จำนวนเซลล์ที่เกิดขึ้นเมื่อระดับจุดเริ่มเปลี่ยนต่ำกว่าระดับ minimum threshold บางครั้งจะมีมากกว่า 171 เซลล์หรือมากกว่า 1 แพ็กเกจ

ดังนั้นเมื่อต้องการให้โครงข่ายมีประสิทธิภาพสูงสุดและมีค่าความเท่าเทียมกันจะต้องตั้งขนาดบัฟเฟอร์และระดับจุดเริ่มเปลี่ยนให้มากเพียงพอตามสมการที่ (5.1) และ (5.2) แต่ในทางปฏิบัติสวิตช์อาจจะมิขนาดบัฟเฟอร์ไม่มากเพียงพอ ถ้าต้องการให้ขนาดบัฟเฟอร์หรือระดับจุดเริ่มเปลี่ยนมากเพียงพอก็จะทำให้เพิ่มต้นทุนให้กับโครงข่ายได้ ฉะนั้นการเลือกใช้ขนาดบัฟเฟอร์หรือการตั้งระดับจุดเริ่มเปลี่ยนจะต้องคำนึงถึงส่วนประกอบต่างๆหลายๆอย่างเข้าด้วยกัน บางครั้งอาจจะเลือกระดับจุดเริ่มเปลี่ยนไม่มากถึงขนาดทำให้ไม่เกิดเซลล์สูญเสียหรือมีประสิทธิภาพสูงสุดแต่จะต้องเลือกขนาดบัฟเฟอร์ที่ไม่ต่ำจนทำให้ค่าความเท่าเทียมกันถึงจุดวิกฤต ตามผลการศึกษาในหัวข้อที่ 5.6.1