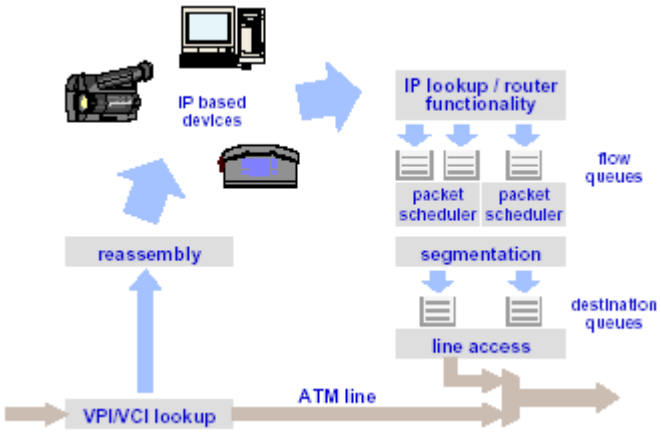
บทที่ 2

การศึกษาค้นคว้าข้อมูล

เครื่องจัดการจราจรในระบบเครือข่าย (Network Traffic Manager) ที่พบเห็นอยู่ในปัจจุบันถูกผลิตขึ้นจากหลากหลายผู้ผลิตโดยใช้สถาปัตยกรรมและเทคโนโลยีที่แตกต่างกัน ซึ่งสถาปัตยกรรมและเทคโนโลยีดังกล่าวล้วนมีข้อดีและข้อด้อยที่แตกต่างกันออกไป ด้วยเหตุนี้ในงานวิจัยนี้จึงได้ทำการศึกษาบทความและงานวิจัยที่เกี่ยวข้องกับเครื่องจัดการจราจรในระบบเครือข่ายเพื่อนำมาประกอบการทำวิจัย โดยจะเน้นในเรื่องอัลกอริทึมและสถาปัตยกรรมที่เลือกใช้

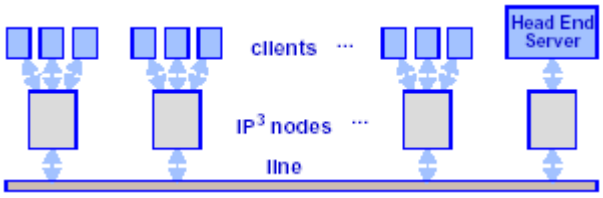
2.1 บทความและงานวิจัยที่เกี่ยวข้องกับเครื่องจัดการจราจรในระบบเครือข่าย

2.1.1 ไอพีแพ็คเกตโปรเซสเซอร์ (IP^3 : IP Packet Processor) โครงงานนี้เกี่ยวข้องกับการออกแบบและสร้างแพ็คเกตโปรเซสเซอร์ขนาดกระทัดรัดบนบอร์ดพีซีไอ (PCI) หนึ่งบอร์ด อัลกอริทึมที่ใช้ในการจัดลำดับแพ็คเกตคือ การจัดเรียงแพ็คเกตตามลำดับความสำคัญแบบไดนามิก (Dynamic Priorities) ภาพรวมการทำงานของไอพีแพ็คเกตโปรเซสเซอร์แสดงดังภาพประกอบ 2-1 ไอพีแพ็คเกตโปรเซสเซอร์จะถูกติดตั้งอยู่ระหว่างเครื่องลูกข่าย (Clients) และเอทีเอ็มแบ็กโบน (ATM Backbone) ซึ่งโทโปโลยีของเครือข่ายแสดงดังภาพประกอบ 2-2



ภาพประกอบ 2-1 การทำงานของไอพีแพ็คเกตโปรเซสเซอร์โดยภาพรวม

(ที่มา : IP^3 Packet Processor: http://www.tik.ee.eth.ch/~ip3/)



ภาพประกอบ 2-2 โทโปโลยีของเครือข่าย

(ที่มา : IP^3 Packet Processor: http://www.tik.ee.eth.ch/~ip3/)

ประเด็นที่น่าสนใจคือวิธีการที่โครงงานนี้นำมาใช้เพื่อเพิ่มความเร็วในการทำงาน ได้แก่

1) การออกแบบอุปกรณ์ที่ต้องการแบนด์วิดท์สูง เช่น โปรเซสเซอร์และคอนโทรลเลอร์ควบคุมการทำงานของหน่วยความจำ (Memory Controller) ด้วย ASIC
2) การลดค่า Latency ที่เกิดขึ้นจากการทำงานของหน่วยความจำโดยการกำหนดตำแหน่งจัดเก็บแพ็คเกตอย่างเหมาะสมและจัดตารางเวลา (Scheduling) การอ่านและเขียนหน่วยความจำ ซึ่งเป็นการใช้ประโยชน์จากการอินเตอร์เฟส SDRAM แบบไปป์ไลน์ เวลาที่ลดลงเป็นเวลาในส่วนของการพรีชาร์จ (Precharge) และการแอคทิเวต (Activate) หน่วยความจำ กล่าวคือ ในการเข้าถึงหน่วยความจำโดยปกติก่อนจะสามารถเข้าถึงหน่วยความจำในแถวถัดไปได้นั้น แถวปัจจุบันจะต้องถูกทำพรีชาร์จเสียก่อน ซึ่งทำให้ต้องเสียเวลาไปส่วนหนึ่งกับกระบวนการนี้ โครงงานนี้จึงนำวิธีการที่เรียกว่า Open-Page Policy และ Closed-Page Policy มาช่วยลดค่า Latency ดังกล่าว โดย Open-Page Policy เป็นการปล่อยให้หน่วยความจำอยู่ในสถานะที่ถูกแอคทิเวตนานที่สุดเท่าที่จะเป็นไปได้เพื่อลดการพรีชาร์จและการแอคทิเวต ส่วนกระบวนการตรงข้ามกันคือ Closed-Page Policy เป็นการพรีชาร์จแถวของหน่วยความจำที่แอคทีพอยู่ทันทีที่สามารถทำได้ มักจะใช้เมื่อมีการเปลี่ยนแถวหรือ Bank บ่อย ๆ
3) การออกแบบโครงสร้างข้อมูล (Data Structure) มาโดยเฉพาะเพื่อลดเวลาของขั้นตอนการทำ Routing Lookup

2.1.2 Algorithm-Architecture Trade-offs in Network Processor Design เป็นวิทยานิพนธ์ที่เกี่ยวข้องกับการออกแบบเน็ตเวิร์กโปรเซสเซอร์ (Network Processor) สำหรับใช้งานกับเครือข่ายที่เชื่อมต่อระหว่างผู้ให้บริการอินเตอร์เน็ต (ISP) กับผู้ใช้บริการและรองรับบริการในเรื่อง

ของคุณภาพการให้บริการ โดยทำการศึกษาฟังก์ชันต่าง ๆ ในการประมวลผลแพ็คเกตและอัลกอริทึมของแต่ละฟังก์ชันที่ใช้งานอยู่ในปัจจุบัน เพื่อหาอัลกอริทึมที่เหมาะสมที่สามารถจัดการในเรื่องของคุณภาพการให้บริการ และหาสถาปัตยกรรมที่เหมาะสมสำหรับเน็ตเวิร์กโปรเซสเซอร์ ซึ่งในการหาอัลกอริทึมและสถาปัตยกรรมดังกล่าว จำเป็นต้องใช้แบบจำลอง (Models) 3 ชนิดได้แก่

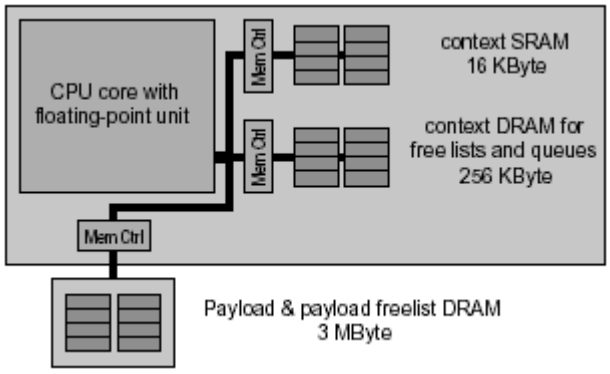
1) แบบจำลองอัลกอริทึม (Algorithm Models) เป็นการกำหนดพฤติกรรมของอัลกอริทึมของฟังก์ชันในการประมวลผลแพ็คเกต
2) แบบจำลองสถาปัตยกรรม (Architecture Models) เป็นการจำลองสถาปัตยกรรมที่จะใช้สร้างเน็ตเวิร์กโปรเซสเซอร์
3) แบบจำลองการสร้างจราจร (Traffic Generation Models) สำหรับจำลองการจราจรที่จะป้อนให้กับเน็ตเวิร์กโปรเซสเซอร์

ในการหาอัลกอริทึมนั้นจะเน้นเฉพาะฟังก์ชัน Policing, การจัดคิว (Queuing), การจัดตารางเวลาแพ็คเกต (Packet Scheduling) เนื่องจากฟังก์ชันอื่น ๆ จะไม่ขึ้นกับอัลกอริทึมที่เลือกใช้

ส่วนสถาปัตยกรรมของเน็ตเวิร์กโปรเซสเซอร์ในปัจจุบันถูกจำแนกตามการประยุกต์ใช้งานหรือตำแหน่งของเน็ตเวิร์กโปรเซสเซอร์ในระบบเครือข่าย ซึ่งในวิทยานิพนธ์นี้ได้จำแนกออกเป็น 3 กลุ่มหลัก ๆ ได้แก่

1) การใช้งานกับแบนด์วิดท์ที่มีปริมาณไม่มากและไม่ต้องการฮาร์ดแวร์พิเศษสำหรับจัดการเรื่องของคุณภาพการให้บริการ เน็ตเวิร์กโปรเซสเซอร์ในกลุ่มนี้ประกอบด้วยซีพียูและฮาร์ดแวร์สำหรับใช้งานฟังก์ชันบางอย่าง
2) การใช้งานกับแบนด์วิดท์ที่มีปริมาณปานกลาง เช่น เครือข่ายอีเทอร์เน็ตความเร็วสูงจนถึงเครือข่ายที่รับส่งข้อมูลที่ความเร็ว 1 กิกะบิตต่อวินาที เน็ตเวิร์กโปรเซสเซอร์ในกลุ่มนี้จะใช้ซีพียูที่ทำงานที่ความเร็วสูงขึ้นกว่ากลุ่มแรก
3) การใช้งานกับแบนด์วิดท์ที่มีปริมาณสูง เช่น เครือข่ายที่รับส่งข้อมูลที่ความเร็วสูงกว่า 1 กิกะบิตต่อวินาที ซึ่งจะเน้นการทำงานในลักษณะขนาน (Parallelism) เป็นสำคัญ

จากการจำลองการทำงานของแบบจำลองต่าง ๆ พบว่า สถาปัตยกรรมของเน็ตเวิร์กโปรเซสเซอร์ที่เหมาะสำหรับเครือข่ายที่เชื่อมต่อระหว่างผู้ให้บริการอินเตอร์เน็ตกับผู้ใช้บริการนั้นประกอบด้วยซีพียู 1 ตัวและหน่วยความจำ 3 ส่วนดังแสดงในภาพประกอบ 2-3



ภาพประกอบ 2-3 สถาปัตยกรรมของเน็ตเวิร์กโปรเซสเซอร์

(ที่มา : Matthias Gries. 2001. Algorithm-Architecture Trade-offs in Network Processor Design, Ph.D. Dissertation, ETH Zurich.)

ข้อสังเกตที่ได้จากวิทยานิพนธ์นี้ ได้แก่ ในการใช้งานกับเครือข่ายความเร็วสูงนั้น วิธีการที่นำมาใช้คือ

1) การใช้ซีพียูที่มีความเร็วสูง
2) การทำงานในลักษณะขนาน
3) การใช้การจำลองการทำงาน (Simulation) ในการหาอัลกอริทึมและสถาปัตยกรรมที่เหมาะสมทำให้มีความยืดหยุ่นในการปรับเปลี่ยนค่าต่าง ๆ

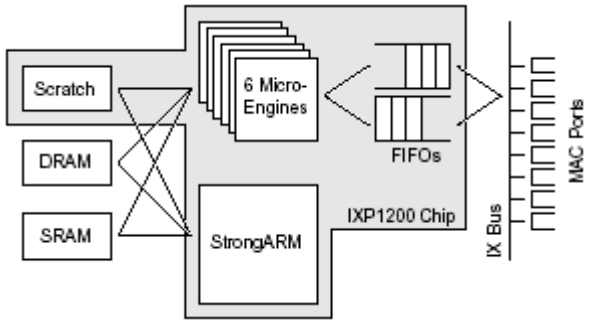
นอกจากนี้การพิจารณาอัลกอริทึมในการจัดคิวนั้น พารามิเตอร์ที่สำคัญคือ

1) ค่า Latency Time
2) Fairness Index
3) ความซับซ้อนในการสร้างฮาร์ดแวร์ให้รองรับการทำงานตามอัลกอริทึมที่เลือกใช้

2.1.3 Evaluating Network Processors in IP Forwarding บทความนี้นำเสนอการหาประสิทธิภาพการทำงานของเน็ตเวิร์กโปรเซสเซอร์ โดยเฉพาะเน็ตเวิร์กโปรเซสเซอร์ในไอพีเราเตอร์ที่ถูกออกแบบมาเพื่อลดผลกระทบของ Latency ซึ่งเกิดขึ้นจากการถ่ายโอนข้อมูลของหน่วยความจำ โดยสามารถส่งต่อไอพีแพ็คเกตขนาดเล็กที่ความเร็ว Line Speed และใช้เทคโนโลยีการออกแบบด้วย ASIC

ส่วนประกอบของบอร์ดต้นแบบ (Prototype Board) ที่นำมาใช้ในการทดสอบเพื่อหาประสิทธิภาพการทำงานของเน็ตเวิร์กโปรเซสเซอร์แสดงดังภาพประกอบ 2-4 โดยประกอบด้วยชิพ

เน็ตเวิร์กโปรเซสเซอร์ของบริษัทอินเทลรุ่น IXP1200 หน่วยความจำชนิด DRAM ขนาด 32 เมกกะไบต์สำหรับเป็นที่พักของแพ็คเกตชั่วคราว และหน่วยความจำชนิด SRAM ขนาด 2 เมกกะไบต์สำหรับเก็บข้อมูล Routing Table รวมทั้ง IX Bus ขนาด 64 บิตและการติดต่อกับบัสแบบพีซีไอ นอกจากนี้ชิพ Media Access Controller (MAC) ภายในชิพ IXP1200 จะมี First-in, First out : FIFO อยู่ 1 คู่เพื่อทำหน้าที่รับและส่งแพ็คเกตระหว่าง IX Bus และเน็ตเวิร์กพอร์ต



ภาพประกอบ 2-4 บล็อกไดอะแกรมของระบบซึ่งใช้เน็ตเวิร์กโปรเซสเซอร์รุ่น IXP1200
(ที่มา : Tammo Spalink, Scott Karlin, Larry Peterson. Evaluating Network Processors in IP Forwarding. Technical Report TR-626-00.)

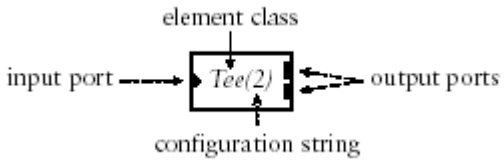
เมื่อได้รับแพ็คเกตเข้ามา ชิพ MAC จะแบ่งแพ็คเกตออกเป็นหน่วยย่อย ๆ เรียกว่า MAC-Packet (MP) ซึ่งมีขนาด 64 ไบต์ จากนั้น MP จะถูกเก็บไว้ในอินพุต FIFO โดยมี μEngine ทำหน้าที่เคลื่อนย้าย MP ของแต่ละแพ็คเกตไปไว้ใน DRAM และเมื่อต้องการจะส่งต่อแพ็คเกต μEngine จะอ่าน MP มาจาก DRAM แล้วจัดเก็บ MP ไว้ในเอาท์พุต FIFO จากนั้นเป็นหน้าที่ของชิพ MAC ที่จะทำหน้าที่จัดส่งแพ็คเกตต่อไป ประสิทธิภาพการทำงานของสถาปัตยกรรมนี้ขึ้นอยู่กับ Latency ที่เกิดขึ้นในการถ่ายโอน MP แต่ละครั้งเป็นสำคัญ ดังแสดงในตาราง 2-1 ซึ่งแสดงค่า Latency ที่วัดได้จากการทำการเคลื่อนย้ายแพ็คเกตในแต่ละคำสั่งของ μEngine

วิธีการลดผลกระทบจาก Latency ซึ่งเกิดขึ้นจากการถ่ายโอนข้อมูลของหน่วยความจำ ซึ่งนำเสนอในบทความนี้คือ การทำงานในลักษณะขนาน แต่ความยุ่งยากจะอยู่ที่การบริหารจัดการกับส่วนประมวลผลย่อย ๆ ให้สามารถใช้งานแบนด์วิดท์ของหน่วยความจำได้อย่างมีประสิทธิภาพสูงสุด

ตาราง 2-1 ค่า Latency ที่วัดได้จากการทำการเคลื่อนย้ายแพ็คเกตในแต่ละคำสั่งของ μEngine (ที่มา : Tammo Spalink, Scott Karlin, Larry Peterson. Evaluating Network Processors in IP Forwarding. Technical Report TR-626-00.)

Instruction	Cycles
FIFO-to-DRAM	45
DRAM-to-FIFO	55
FIFO-to-μEngine	30
μEngine-to-FIFO	40
DRAM-to-μEngine	40
μEngine-to-DRAM	50
SRAM-to-μEngine	30
μEngine-to-SRAM	30
Scratch-to-μEngine	25
μEngine-to-Scratch	25

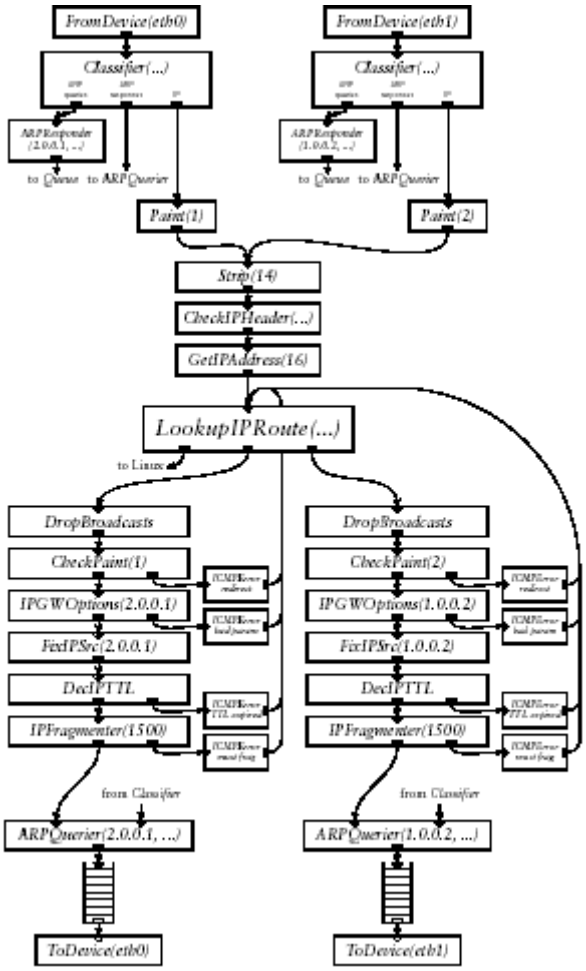
2.1.4 The Click Modular Router วิทยานิพนธ์นี้เกี่ยวข้องกับการสร้างเราเตอร์หรือแพ็คเกตโปรเซสเซอร์ที่มีความยืดหยุ่นและสามารถปรับแต่งค่าต่าง ๆ ได้ โดยใช้ซอฟท์แวร์ที่ชื่อว่า คลิก (Click) การสร้างคลิกเราเตอร์นั้นสร้างขึ้นมาจากโมดูลประมวลผลแพ็คเกตย่อย ๆ ที่เรียกว่า อีลีเมนต์ (Element) ดังแสดงในภาพประกอบ 2-5 แต่ละอีลีเมนต์จะถูกนำมาสร้างเป็นฟังก์ชันการทำงานต่าง ๆ ของเราเตอร์เช่น การจำแนกแพ็คเกต การจัดคิว และการติดต่อกับอุปกรณ์ในเครือข่าย เป็นต้น



ภาพประกอบ 2-5 ตัวอย่างของอีลีเมนต์
(ที่มา : Eddie Kohler. 2001. “The Click Modular Router”. PhD thesis, Massachusetts Institute of Technology.)

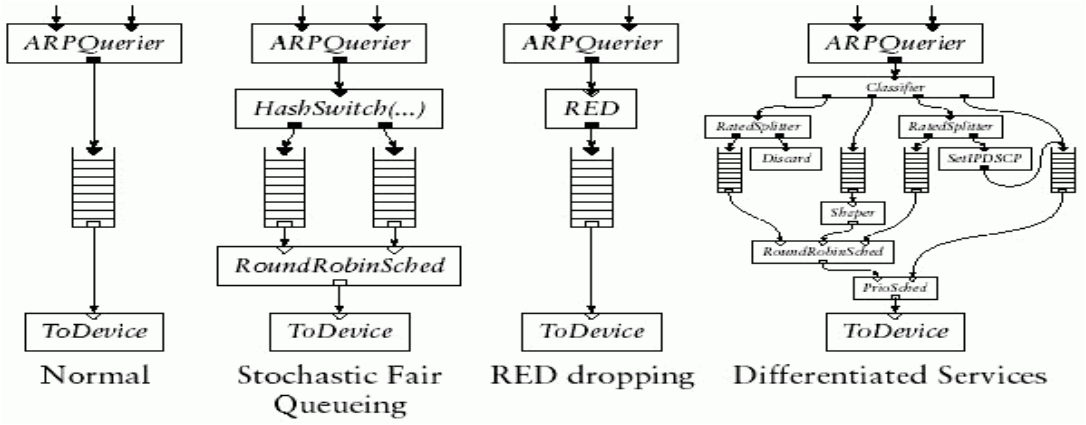
สถาปัตยกรรมแบบคลิกจะเน้นความสนใจอยู่ที่อีลีเมนต์ โดยอีลีเมนต์จะถูกแทนด้วยกล่องสี่เหลี่ยมและเชื่อมต่อกันด้วยลูกศร แพ็คเกตจะถูกส่งจากอีลีเมนต์หนึ่งไปยังอีกอีลีเมนต์หนึ่งผ่านทางลูกศร ทิศทางของลูกศรบ่งบอกถึงทิศทางการเคลื่อนที่ของแพ็คเกต การเชื่อมต่ออีลีเมนต์ในลักษณะนี้เรียกว่า Directed Graph โดยที่แต่ละอีลีเมนต์เป็นออบเจ็คที่เขียนด้วยภาษา C++

ตัวอย่างของไอพีเราเตอร์ที่เขียนขึ้นด้วยคลิกซอฟท์แวร์แสดงดังภาพประกอบ 2-6 ซึ่งไอพีเราเตอร์ดังกล่าวประกอบด้วยการ์ดแลน 2 การ์ดโดยทำการส่งต่อแพ็คเกตประเภทยูนิแคส (Unicast) หากต้องการเพิ่มฟังก์ชันต่าง ๆ ให้กับไอพีเราเตอร์นั้นทำได้ง่ายเนื่องจากคลิกเราเตอร์มีลักษณะเป็นโมดูล ทำให้การเพิ่มเติมฟังก์ชันการทำงานต่าง ๆ ทำได้โดยการเขียนฟังก์ชันเพิ่มเติมขึ้นมาจากอีลีเมนต์ที่มีอยู่ โดยฟังก์ชันการทำงานที่เขียนเพิ่มขึ้นนี้จะมีผลแค่ต่อการจัดเก็บแพ็คเกตของเราเตอร์เท่านั้น แต่จะไม่ส่งผลกระทบต่อการหาเส้นทางของแพ็คเกต (Routing Decision) ของเราเตอร์ที่มีอยู่เดิม ดังภาพประกอบ 2-7



ภาพประกอบ 2-6 ตัวอย่างของไอพีเราเตอร์

(ที่มา : Eddie Kohler. 2001. “The Click Modular Router”. PhD thesis, Massachusetts Institute of Technology.)



ภาพประกอบ 2-7 การจัดคิวแบบต่าง ๆ ของไอพีเราเตอร์

(ที่มา : Eddie Kohler. 2001. “The Click Modular Router”. PhD thesis, Massachusetts Institute of Technology.)

การหาประสิทธิภาพของเราเตอร์ที่สร้างขึ้นด้วยคลิกซอฟท์แวร์นั้นทำโดยการใช้พีซี 9 เครื่องรันบนระบบปฏิบัติการลีนุกซ์เวอร์ชัน 2.2.16 ที่ได้รับการดัดแปลงเพื่อให้สามารถรองรับการทำโพลลิ่ง (Polling) ของคลิกซอฟท์แวร์ซึ่งประกอบด้วยพีซีเราเตอร์ 1 เครื่อง เครื่องต้นทาง 4 เครื่องและเครื่องปลายทาง 4 เครื่อง พีซีเราเตอร์ประกอบด้วยซีพียูเพนเทียมทรี ความเร็ว 700 เมกกะเฮิร์ทบนเมนบอร์ดรุ่น L440GX+ และการ์ดแลนความเร็ว 100 เมกกะบิตต่อวินาที ซึ่งใช้ชิพอีเทอร์เน็ตคอนโทรลเลอร์รุ่น 21140 Tulip ส่วนพีซีอื่น ๆ ใช้ซีพียูเพนเทียมทรี 733 เมกกะเฮิร์ทและเพนเทียมโปรความเร็ว 200 เมกกะเฮิร์ทตามลำดับ ซึ่งเชื่อมต่อกับพีซีเราเตอร์ในลักษณะแบบ Point-to-Point และรับส่งข้อมูลแบบฟูลดูเพล็กซ์ที่ความเร็ว 100 เมกกะบิตต่อวินาที โดยทำการส่งยูดีพี (User Datagram Protocol : UDP) แพ็คเกตจากเครื่องต้นทางผ่านเราเตอร์ไปยังเครื่องปลายทาง และพิจารณาอัตราการส่งต่อแพ็คเกตของเราเตอร์ที่อัตราขาเข้าของแพ็คเกตที่แตกต่างกัน

หลังจากวิเคราะห์ประสิทธิภาพการทำงานพบว่าคลิกเราเตอร์มีประสิทธิภาพใกล้เคียงกับการใช้พีซีเราเตอร์อื่น ๆ รวมทั้งจากการใช้เครื่องมือที่มีประสิทธิภาพทำให้ช่วยลดเวลาการประมวลผลของซีพียูต่อแพ็คเกตของคลิกไอพีเราเตอร์ลงได้ถึง 34% ส่งผลให้อัตราเร็วในการส่งต่อแพ็คเกตขนาด 64 ไบต์ของคลิกไอพีเราเตอร์ ที่ถูกออปติไมซ์แล้วมีค่าเท่ากับ 400,000 แพ็คเกตต่อวินาที

ประเด็นที่น่าสนใจของวิทยานิพนธ์นี้คือคลิกเราเตอร์เป็นพีซีเราเตอร์หรือเราเตอร์ประเภท Software-Based ซึ่งข้อดีของเราเตอร์ประเภทนี้อยู่ที่ความยืดหยุ่นในการปรับเปลี่ยนค่าต่าง ๆ โดยขีดความสามารถของเราเตอร์ขึ้นอยู่กับประสิทธิภาพของพีซีและซอฟท์แวร์ที่ใช้ การนำไปใช้งาน

กับระบบเครือข่ายความเร็วสูงนั้นเราเตอร์ประเภท Software-Based อาจประมวลผลได้ไม่ทัน เนื่องจากประสบปัญหาคอขวดจากการเข้าถึงหน่วยความจำของระบบ

2.1.5 Network Processors: The Optimal Building Block for Next Generation IP Routers เอกสารทางเทคนิคฉบับนี้นำเสนอรูปแบบของสถาปัตยกรรมของเอดจ์เราเตอร์ (Edge Router) ยุคใหม่ซึ่งประกอบด้วย

1) เราเตอร์ประเภท Software-Based
2) เราเตอร์ประเภท ASIC-Based
3) เราเตอร์ประเภท Network Processor-Based

ประเด็นที่ใช้ในการเปรียบเทียบข้อดีและข้อเสียของแต่ละสถาปัตยกรรมคือ

1) มีจำนวนพอร์ตมากกว่า
2) มี Throughput มากขึ้น
3) มีความสามารถในการวิเคราะห์แพ็คเกตได้ละเอียดมากขึ้น
4) มีความยืดหยุ่นในการปรับเปลี่ยนให้เข้ากับการเปลี่ยนแปลงของมาตรฐานเชิงอุตสาหกรรมและสามารถเพิ่มเติมการให้บริการใหม่ ๆ ได้ง่าย

ตาราง 2-2 การเปรียบเทียบสถาปัตยกรรมของเราเตอร์

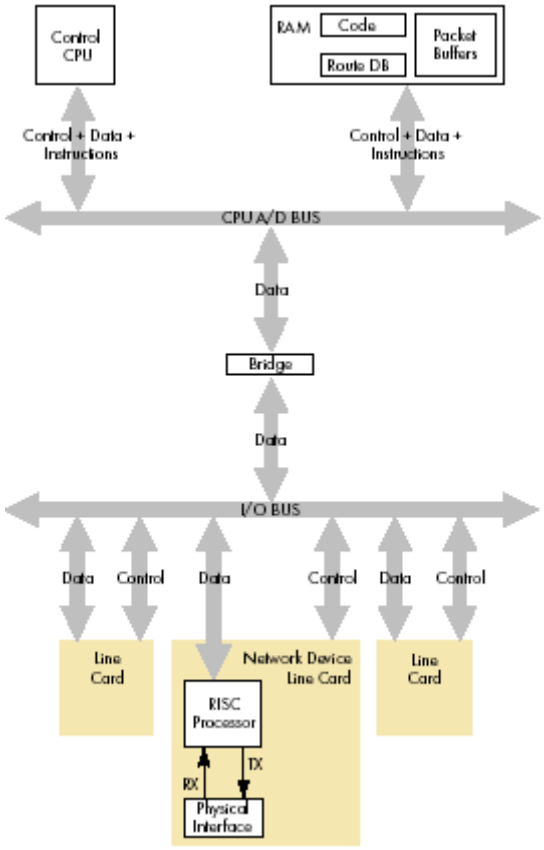
(ที่มา : Alcatel. Network Processors: The Optimal Building Block for Next Generation IP Router. Technical Paper. November 2000.)

	Software-based Routers	ASIC-based Routers	Network Processor-based Routers
Density	X	X	X
Performance/ throughput		X	X
Intelligence/ QoS (for high traffic volumes and without impacting performance)	X		X
Ease of provisioning new services and adapting to changing standards	X		X

เมื่อเปรียบเทียบสถาปัตยกรรมทั้ง 3 ประเภทในแง่ต่าง ๆ ดังแสดงในตาราง 2-2 พบว่ามีเพียงเราเตอร์ประเภท Network Processor-Based เท่านั้นที่มีคุณสมบัติครบถ้วนสำหรับเราเตอร์ยุคใหม่ รายละเอียดของเราเตอร์แต่ละประเภทมีดังนี้คือ

2.1.5.1 เราเตอร์ประเภท Software-Based เราเตอร์ประเภทนี้จะถูกมองว่าเป็นแอพพลิเคชันซึ่งรันอยู่บนเครื่องคอมพิวเตอร์ที่ใช้งานทั่วไป เราเตอร์ประเภท Software-Based ที่ผลิตขึ้นมาเพื่อการค้านั้นมีลักษณะเหมือนกันคือใช้ซีพียูประเภทที่ใช้งานทั่วไปและมีโปรแกรมทำหน้าที่เลือกเส้นทางและส่งต่อแพ็คเกต ฮาร์ดแวร์ที่ใช้มักจะเป็นฮาร์ดแวร์ซึ่งใช้งานอยู่ในเครื่องคอมพิวเตอร์ทั่วไป เราเตอร์ประเภทนี้ไม่มีฮาร์ดแวร์เพิ่มเติมที่นำมาช่วยในการทำงานที่นอกเหนือจากการทำ Encapsulation และ De-Encapsulation ในชั้น Physical หรือ Data link Layer

ข้อดีของเราเตอร์ประเภทนี้คือ การนำเสนอคุณลักษณะใหม่ ๆ และการแก้ไขข้อผิดพลาดสามารถทำได้โดยการอัพเกรดซอฟท์แวร์ ซึ่งช่วยลดการลงทุนในด้านฮาร์ดแวร์ลง



ภาพประกอบ 2-8 สถาปัตยกรรมของเราเตอร์ประเภท Software-Based

(ที่มา : Alcatel. Network Processors: The Optimal Building Block for Next Generation IP Router. Technical Paper. November 2000.)

ข้อเสียของเราเตอร์ประเภทนี้คือซีพียูที่ใช้เป็นซีพียูสำหรับใช้งานทั่วไปซึ่งไม่ได้ถูกออปติไมซ์สำหรับฟังก์ชันการทำงานของเราเตอร์ยุคใหม่หรือการเคลื่อนย้ายข้อมูลอินพุตและเอาท์พุตจำนวนมากเนื่องจากซีพียูทั่วไปจะใช้บัสเดียวกันระหว่างสัญญาณข้อมูลและสัญญาณควบคุม รวมทั้งการที่ส่วนควบคุม ส่วนข้อมูล และส่วนการจัดการไม่ได้แยกจากกันเป็นการจำกัดเราเตอร์ประเภทนี้ทำให้ไม่สามารถปรับเปลี่ยนเพื่อรองรับระบบเครือข่ายขนาดใหญ่ได้

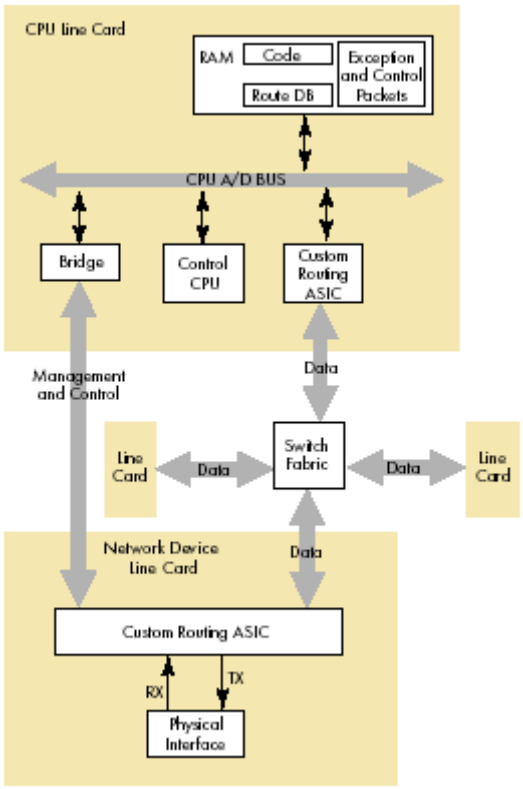
หากพิจารณาจากภาพประกอบ 2-8 พบว่าสถาปัตยกรรมนี้มีปัญหาคอขวดอยู่ที่ A/D บัสของซีพียู และแม้ว่าอุปกรณ์บนไลน์การ์ด (Line Card) จะมีซีพียูต่างหากที่ช่วยในการหาเส้นทางให้แพ็คเกต แต่การรับส่งข้อมูลและการควบคุมการทำงานก็ยังคงเป็นภาระที่หนักสำหรับซีพียูในการประมวลผลและการใช้ทรัพยากรของซีพียู

2.1.5.2 เราเตอร์ประเภท ASIC-Based เราเตอร์ประเภทนี้เป็นการผสมผสานกันระหว่างสถาปัตยกรรมแบบ RISC และ CISC เพื่อเป็นการปรับภาระงานให้สมดุล โดยปกติแล้ว ASIC จะจัดการในส่วนข้อมูลและส่วนควบคุมบางส่วน เช่น การจำแนกแพ็คเกต การกรองและการส่งต่อแพ็คเกต ในขณะที่ซีพียูดูแลในส่วนของการจัดการและส่วนควบคุมอื่น ๆ รวมถึงการให้บริการต่าง ๆ ที่เพิ่มมูลค่าให้กับระบบ เช่น การเก็บข้อมูลทางสถิติ เราเตอร์ประเภทนี้สามารถปรับเปลี่ยนให้รองรับระบบเครือข่ายขนาดใหญ่ได้ดีรวมทั้งส่วนข้อมูลและส่วนควบคุมจะถูกแยกกันอย่างมีประสิทธิภาพ

การใช้เราเตอร์ประเภท ASIC-Based เหมาะกับการใช้งานเป็นคอร์เราเตอร์ (Core Router) ซึ่งทำหน้าที่ในการส่งต่อแพ็คเกตจำนวนมาก

ปัญหาของเราเตอร์ประเภทนี้เกี่ยวข้องกับการเพิ่มคุณลักษณะใหม่ ๆ ของระบบ ซึ่งหมายถึงการเปลี่ยนแปลงฮาร์ดแวร์และมีค่าใช้จ่ายที่สูงขึ้น นอกจากนี้เราเตอร์ประเภท ASIC-Based ยังใช้เวลาในการออกแบบนานกว่าเราเตอร์ประเภท Software-Based มาก

สถาปัตยกรรมของเราเตอร์ประเภท ASIC-Based ในภาพประกอบ 2-9 นั้นอุปกรณ์ ASIC ต่อตรงเข้ากับเส้นทางรับส่งข้อมูลของอุปกรณ์สวิตช์เลือกเส้นทาง และอุปกรณ์ ASIC อีกส่วนต่อตรงเข้ากับอุปกรณ์อินเตอร์เฟสภายนอกโดยอาจจะทำฟังก์ชัน Segmentation และ Reassembly หรือที่เรียกว่า SAR หรือฟังก์ชันอื่น ๆ ส่วนซีพียูดูแลเรื่องการเซ็ตค่าคอนฟิกกูเรชันและการควบคุม ASIC รวมถึงการเก็บข้อมูลทางสถิติ การอินเตอร์รัพ การจัดการข้อมูลในการเลือกเส้นทางและอื่น ๆ



ภาพประกอบ 2-9 สถาปัตยกรรมของเราเตอร์ประเภท ASIC-Based

(ที่มา : Alcatel. Network Processors: The Optimal Building Block for Next Generation IP Router. Technical Paper. November 2000.)

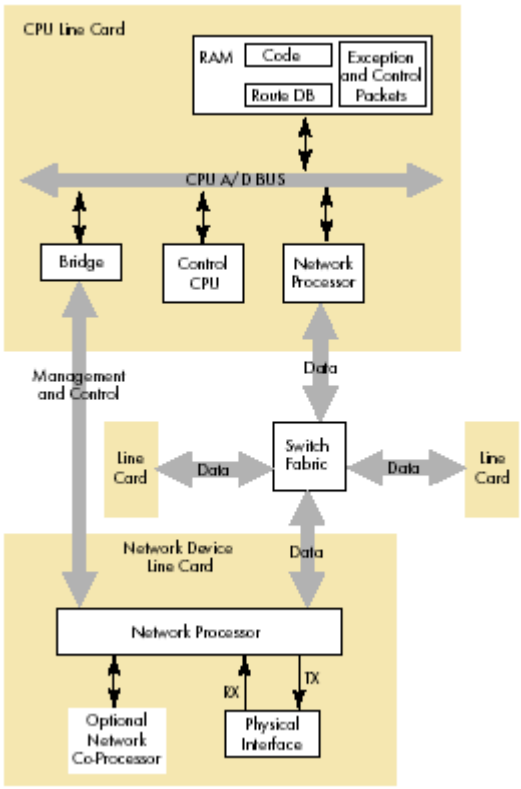
2.1.5.3 เราเตอร์ประเภท Network Processor-Based เป็นการนำความสามารถในเรื่องความเร็วของ ASIC ผสมผสานกับความยืดหยุ่นในการใช้ซีพียูทั่วไปเพื่อให้สามารถรองรับความต้องการของเราเตอร์ยุคใหม่

เราเตอร์ประเภทนี้เป็นอุปกรณ์ที่ถูกออปติไมซ์มาเพื่อเพิ่มความเร็วในการประมวลผลแพ็คเกต คุณสมบัติของเราเตอร์ประเภทนี้คือ มีแบนด์วิดท์ของอินพุตและเอาท์พุตที่กว้าง และสามารถโปรแกรมได้ซึ่งเป็นคุณสมบัติที่ทำให้เราเตอร์ประเภทนี้เหนือกว่า เราเตอร์ประเภท ASIC-Based โดยใช้ภาษาเครื่องในการควบคุมการทำงานของฮาร์ดแวร์รวมทั้งยังสามารถพัฒนาอัลกอริทึมสำหรับตรวจสอบอัตราเร็วในการรับส่งข้อมูล ควบคุมการไหลของแพ็คเกตและปรับแต่งการจราจรในระบบเครือข่ายโดยใช้ซอฟท์แวร์ สำหรับเราเตอร์ประเภทนี้บัสข้อมูลที่ใช้ในการรับส่งข้อมูลกับระบบเครือข่ายจะแยกจากบัสข้อมูลและบัสควบคุมของซีพียู ทำให้มีการแยกกันของอินพุตเอาท์พุตและบัสในการคำนวณทำให้สามารถใช้ขีดความสามารถของทั้ง ASIC และความ

ยืดหยุ่นของซอฟท์แวร์ได้อย่างเต็มที่ นอกจากนี้ภายในเน็ตเวิร์กโปรเซสเซอร์ยังประกอบด้วยกลุ่มของหน่วยประมวลผลย่อย ๆ เพื่อกระจายภาระการประมวลผลและมีการประมวลผลแบบขนาน

สถาปัตยกรรมของเราเตอร์ประเภท Network Processor-Based แสดงดังภาพประกอบ 2-10 ซึ่งจะเห็นว่ามีความคล้ายคลึงกับสถาปัตยกรรมของเราเตอร์ประเภท ASIC-Based มาก โดยที่เน็ตเวิร์กโปรเซสเซอร์จะต่อตรงเข้ากับบัสข้อมูลของอุปกรณ์สวิตช์ และเน็ตเวิร์กโปรเซสเซอร์อีกส่วนจะต่ออยู่กับอุปกรณ์อินเตอร์เฟสภายนอก ซีพียูซึ่งแยกออกมาต่างหากทำหน้าที่เซ็ตค่าคอนฟิกกูเรชันและการควบคุมการทำงานของเน็ตเวิร์กโปรเซสเซอร์

ในทางปฏิบัติมักจะต้องการแยกแต่ละฟังก์ชันการทำงานและส่วนของภาครับและส่งออกจากกันเพื่อให้ทั้งภาครับและส่งสามารถทำงานอย่างอิสระจากกัน



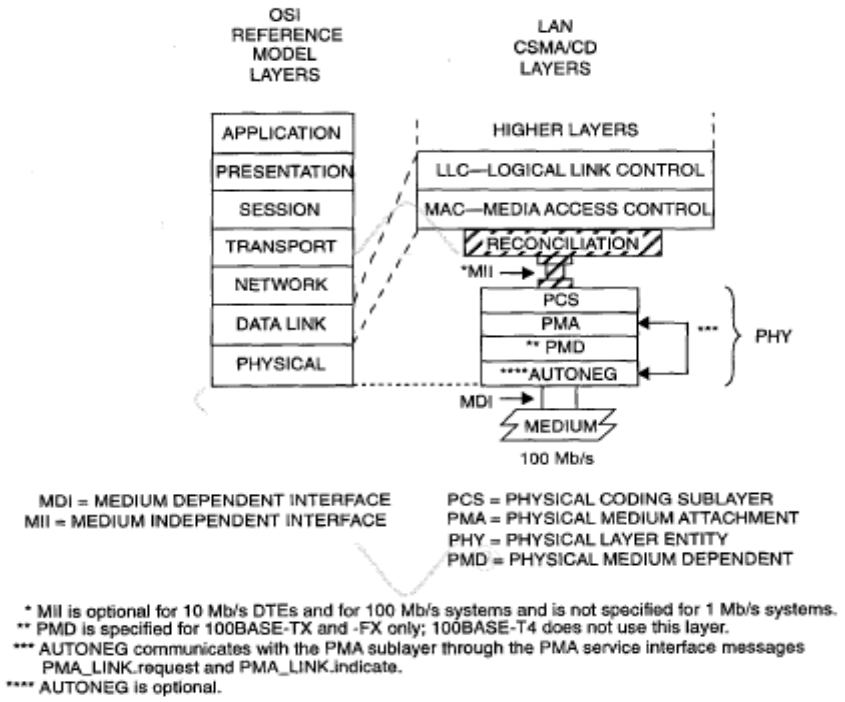
ภาพประกอบ 2-10 สถาปัตยกรรมของเราเตอร์ประเภท Network Processor-Based

(ที่มา : Alcatel. Network Processors: The Optimal Building Block for Next Generation IP Router. Technical Paper. November 2000.)

เน็ตเวิร์กโปรเซสเซอร์เป็นสถาปัตยกรรมที่เหมาะสมสำหรับการใช้เป็นเอดจ์เราเตอร์ ซึ่งเป็นการรวมข้อดีของสถาปัตยกรรมแบบที่ใช้ ASIC และแบบซอฟท์แวร์ เน็ตเวิร์กโปรเซสเซอร์มีข้อดีเหนือกว่าสถาปัตยกรรม RISC ทั่วไปเนื่องจากมีส่วนอินพุตและเอาท์พุตที่มีแบนด์วิดท์ที่กว้างเพียงพอกับความต้องการของระบบเครือข่าย และมีข้อดีเหนือกว่าสถาปัตยกรรมแบบที่ใช้ ASIC ตรงที่สามารถต่อตรงเข้ากับส่วนอินเตอร์เฟสของอุปกรณ์เครือข่ายความเร็วสูงทำให้สามารถตรวจสอบแพ็คเกตได้อย่างละเอียดที่ความเร็ว Wire Speed ซึ่งเป็นไปไม่ได้สำหรับสถาปัตยกรรมแบบที่ใช้ ASIC

2.2 บทความและทฤษฎีที่เกี่ยวข้อง

2.2.1 Media Independent Interface หรือ MII คือกลุ่มของอุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เชื่อมต่อส่วน Medium Access Control (MAC) Sublayer ที่อยู่ในอุปกรณ์เครือข่ายเข้ากับอุปกรณ์ใน Physical Layer หรือที่เรียกว่า PHY ดังแสดงในภาพประกอบ 2-11



ภาพประกอบ 2-11 ตำแหน่งของ MII ใน Protocol Stack

(ที่มา : IEEE Std 802.3u-1995)

ซึ่งคุณสมบัติและการเชื่อมต่อต่าง ๆ ระบุไว้ในมาตรฐานของ IEEE Std 802.3u-1995 คุณสมบัติที่สำคัญของ MII มีดังนี้คือ

1) สามารถรองรับอัตราเร็วในการรับส่งข้อมูลที่ 10 และ 100 เมกกะบิตต่อวินาที
2) สัญญาณข้อมูลและ Delimiter จะซิงโครไนซ์เข้ากับสัญญาณนาฬิกา
3) สายสัญญาณข้อมูลขนาด 4 บิตของภาครับและส่งแยกอิสระจากกัน
4) ระดับของสัญญาณที่ใช้ติดต่อเป็นแบบทีทีแอลและสามารถติดต่อกับสัญญาณดิจิตอลของ CMOS ASIC ทั่วไปได้
5) มีการอินเตอร์เฟสในส่วนของการควบคุมที่ง่าย
6) สามารถขับสัญญาณให้สายเคเบิลได้จำกัดที่ความยาวค่าหนึ่ง

MII ถูกออกแบบมาเพื่อทำให้ชิพอีเทอร์เน็ตในอุปกรณ์เครือข่ายไม่จำเป็นต้องรับรู้ถึงความแตกต่างในการส่งสัญญาณของการใช้งานสื่อกลางในการรับส่งข้อมูลที่แตกต่างกัน โดยการแปลงสัญญาณที่ได้รับมาด้วยอุปกรณ์ที่เรียกว่า Transceiver ให้อยู่ในรูปแบบของสัญญาณดิจิตอลก่อนส่งต่อให้ชิพอีเทอร์เน็ตซึ่งอยู่ในอุปกรณ์เครือข่าย

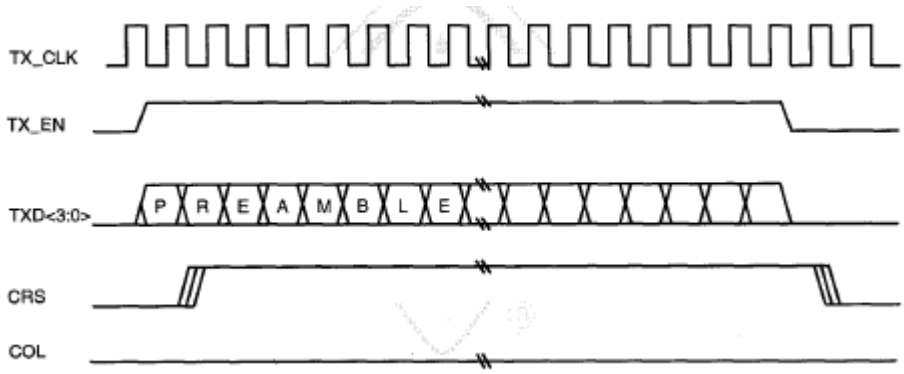
2.2.1.1 ข้อกำหนดของสัญญาณของ MII

2.2.1.1.1 TX_CLK (Transmit Clock) เป็นสัญญาณนาฬิกาต่อเนื่องซึ่งใช้เป็นสัญญาณอ้างอิงสำหรับการส่งสัญญาณ TX_EN, TXD และ TX_ER จาก Reconciliation Sublayer ไปยัง PHY โดยสัญญาณ TX_CLK ส่งมาจาก PHY

2.2.1.1.2 RX_CLK (Receive Clock) เป็นสัญญาณนาฬิกาต่อเนื่องซึ่งใช้เป็นสัญญาณอ้างอิงสำหรับการส่งสัญญาณ RX_DV, RXD และ RX_ER จาก PHY ไปยัง Reconciliation Sublayer โดยสัญญาณ RX_CLK ส่งมาจาก PHY

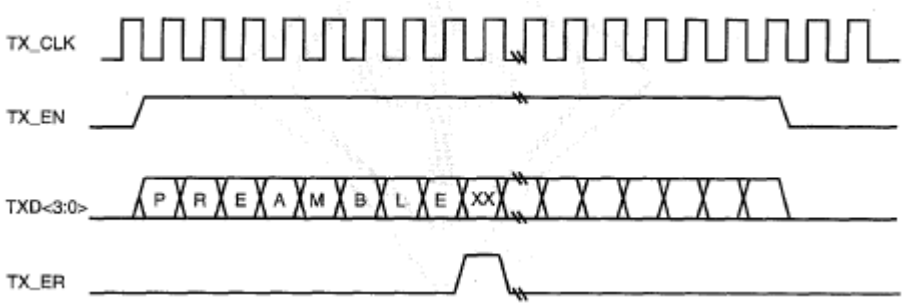
2.2.1.1.3 TX_EN (Transmit Enable) เป็นสัญญาณที่บอกให้ทราบว่า Reconciliation Sublayer กำลังส่งข้อมูลผ่านทาง MII โดยสัญญาณ TX_EN ซึ่งสร้างโดย Reconciliation Sublayer จะแอคทีพตลอดการส่งข้อมูลและซิงโครไนซ์เข้ากับข้อมูล 4 บิต (Nibble) แรกของ Preamble ดังแสดงในภาพประกอบ 2-12

2.2.1.1.4 TXD (Transmit Data) เป็นกลุ่มของสัญญาณข้อมูล 4 เส้นคือ TXD<3:0> ซึ่งส่งมาจาก Reconciliation Sublayer โดยมี TXD<0> เป็นบิตต่ำสุด สัญญาณ TXD <3:0> จะซิงโครไนซ์เข้ากับสัญญาณ TX_CLK ตัวอย่างการส่งข้อมูลของสัญญาณ TXD แสดงดังภาพประกอบ 2-12



ภาพประกอบ 2-12 การส่งข้อมูลเมื่อไม่มีการชนกันของข้อมูลเกิดขึ้น
(ที่มา : IEEE Std 802.3u-1995)

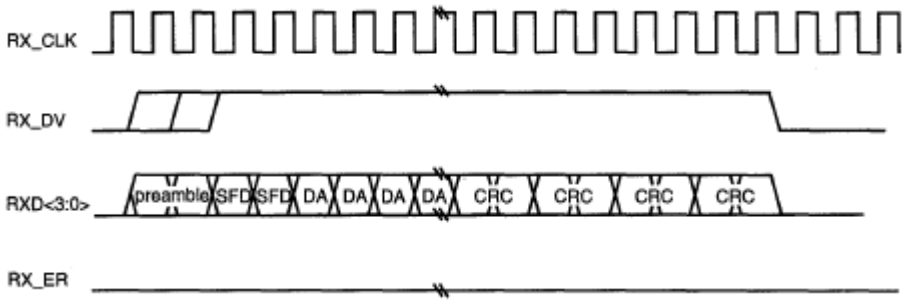
2.2.1.1.5 TX_ER (Transmit Coding Error) การส่งสัญญาณ TX_ER จะซิงโครไนซ์เข้ากับสัญญาณ TX_CLK เมื่อสัญญาณ TX_ER แอคทีพเป็นเวลา 1 คาบสัญญาณ TX_CLK หรือมากกว่านั้น ในขณะที่ TX_EN แอคทีพ PHY จะส่งสัญญาณบางอย่างที่ไม่ใช่ข้อมูลที่ถูกต้องไปกับเฟรมข้อมูล ตัวอย่างการส่งข้อมูลที่มีความผิดพลาดในการส่งข้อมูลเกิดขึ้นแสดงดังภาพประกอบ 2-13 การแอคทีพของสัญญาณ TX_ER จะไม่มีผลกับการส่งข้อมูลหาก PHY ส่งข้อมูลที่อัตราเร็ว 10 เมกกะบิตต่อวินาที



ภาพประกอบ 2-13 การส่งข้อมูลที่มีความผิดพลาดเกิดขึ้น
(ที่มา : IEEE Std 802.3u-1995)

2.2.1.1.6 RX_DV (Receive Data Valid) เป็นสัญญาณที่สร้างโดย PHY เพื่อบอกให้ทราบว่า PHY กำลังส่งสัญญาณข้อมูล RXD<3:0> และสัญญาณดังกล่าวซิงโครไนซ์เข้ากับสัญญาณ RX_CLK การที่ Reconciliation Sublayer และ MAC Sublayer จะสามารถรับ

ข้อมูลได้อย่างถูกต้องนั้นสัญญาณ RX_DV จะต้องครอบคลุมเฟรมข้อมูลโดยจะต้องแอคทีพก่อน Start Frame Delimiter (SFD) และสิ้นสุดก่อน End-of-Frame Delimiter (EFD) ดังภาพประกอบ 2-14

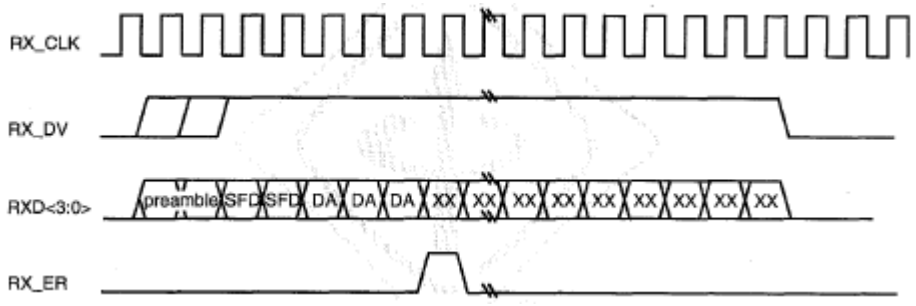


ภาพประกอบ 2-14 การรับข้อมูลที่ไม่มีความผิดพลาดเกิดขึ้น

(ที่มา : IEEE Std 802.3u-1995)

2.2.1.1.7 RXD (Receive Data) เป็นกลุ่มของสัญญาณข้อมูล 4 เส้นคือ RXD<3:0> ซึ่งส่งมาจาก PHY โดยมี RXD<0> เป็นบิตต่ำสุด สัญญาณ RXD<3:0> จะซิงโครไนซ์เข้ากับสัญญาณ RX_CLK

2.2.1.1.8 RX_ER (Receive Error) เป็นสัญญาณที่ส่งจาก PHY โดยสัญญาณ RX_ER จะแอคทีพเป็นเวลา 1 คาบสัญญาณ RX_CLK หรือมากกว่านั้นเพื่อบอก Reconciliation Sublayer ว่า PHY ตรวจพบความผิดพลาดของข้อมูลที่ส่งจาก PHY ไปยัง Reconciliation Sublayer โดยสัญญาณ RX_ER จะซิงโครไนซ์เข้ากับสัญญาณ RX_CLK การรับข้อมูลที่มีความผิดพลาดเกิดขึ้นแสดงดังภาพประกอบ 2-15

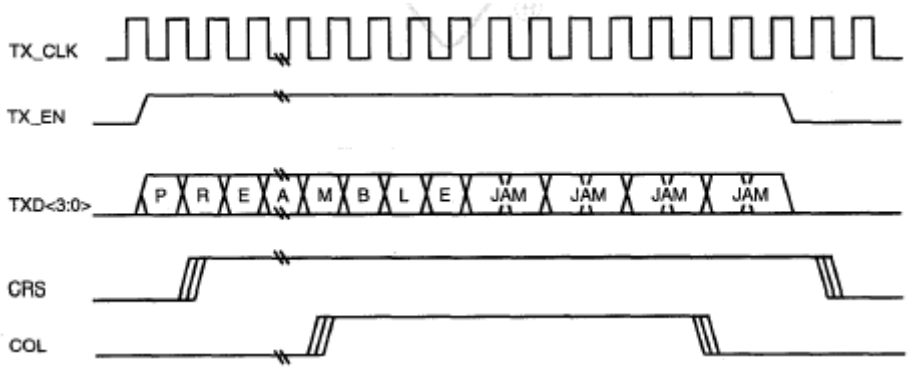


ภาพประกอบ 2-15 การรับข้อมูลที่มีความผิดพลาดเกิดขึ้น

(ที่มา : IEEE Std 802.3u-1995)

2.2.1.1.9 CRS (Carrier Sense) จะถูกทำให้แอคทีพโดย PHY เมื่อเส้นทางรับส่งข้อมูลไม่ว่างหรือมีการรับหรือส่งข้อมูลเกิดขึ้น และในกรณีที่มีการชนกันของข้อมูลเกิดขึ้นนั้น สัญญาณ CRS จะแอคทีพตลอดช่วงเวลาที่มีการชนกันของข้อมูล สัญญาณ CRS ไม่จำเป็นต้องซิงโครไนซ์เข้ากับสัญญาณ TX_CLK หรือ RX_CLK

2.2.1.1.10 COL (Collision Detected) จะถูกทำให้แอคทีพโดย PHY เมื่อตรวจพบว่ามีการชนกันของข้อมูลเกิดขึ้นและแอคทีพไปตลอดในขณะที่ยังมีการชนกันของข้อมูล แสดงดังภาพประกอบ 2-16 สัญญาณ COL ไม่จำเป็นต้องซิงโครไนซ์เข้ากับสัญญาณ TX_CLK หรือ RX_CLK



ภาพประกอบ 2-16 การส่งข้อมูลที่มีการชนกันของข้อมูลเกิดขึ้น
(ที่มา : IEEE Std 802.3u-1995)

2.2.1.1.11 MDC (Management Data Clock) สร้างโดย Station Management Entity ส่งไปยัง PHY เพื่อเป็นสัญญาณอ้างอิงในการส่งข้อมูลของสัญญาณ MDIO สัญญาณ MDC เป็นสัญญาณประเภท Aperiodic และมีคาบต่ำสุดเท่ากับ 400 นาโนวินาที โดยไม่ขึ้นกับคาบของสัญญาณ TX_CLK และ RX_CLK

2.2.1.1.12 MDIO (Management Data Input/Output) เป็นสัญญาณแบบ 2 ทิศทางซึ่งส่งไปมาระหว่าง PHY และ STA โดย STA ใช้สัญญาณ MDIO ในการส่งสัญญาณควบคุมซึ่งจะซิงโครไนซ์เข้ากับสัญญาณ MDC ส่วน PHY ใช้สัญญาณ MDIO ในการส่งข้อมูลสถานะต่าง ๆ

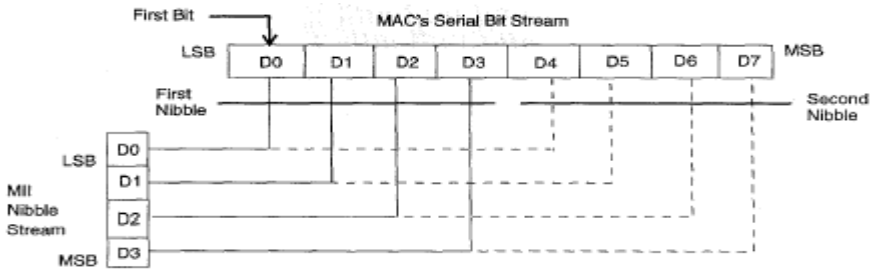
2.2.1.2 โครงสร้างของเฟรมข้อมูลของ MII

ข้อมูลที่ส่งผ่าน MII จะมีรูปแบบดังแสดงในภาพประกอบ 2-17 ซึ่งการรับและส่งข้อมูลของ MII นั้นจะรับส่งข้อมูลครั้งละ 4 บิตโดยมีลำดับการรับและส่งข้อมูลดังแสดงในภาพประกอบ 2-18 ซึ่งข้อมูล 4 บิตล่างคือ บิต 0 ถึง 3 จะถูกส่งออกไปก่อนข้อมูล 4 บิตบน

<inter-frame><preamble><sfd><data><efd>

ภาพประกอบ 2-17 รูปแบบเฟรมข้อมูลของ MII

(ที่มา : IEEE Std 802.3u-1995)



ภาพประกอบ 2-18 ลำดับการรับและส่งข้อมูล

(ที่มา : IEEE Std 802.3u-1995)

2.2.1.2.1 Inter-frame เป็นช่วงเวลาหนึ่งระหว่างไม่มีการรับส่งข้อมูลบน MII ซึ่งสามารถดูได้จากสถานะไม่แอคทีพของสัญญาณ RX_DV ของภาครับและสัญญาณ TX_EN ของภาคส่งของ MII

2.2.1.2.2 Preamble และ Start of Frame Delimiter

2.2.1.2.2.1 การส่งข้อมูล ก่อนการส่งข้อมูล Preamble ซึ่งประกอบด้วยบิตข้อมูลจำนวน 7 Octet ดังนี้คือ 10101010 10101010 10101010 10101010 10101010 10101010 10101010 จะถูกส่งออกไปก่อนในลักษณะการส่งแบบอนุกรม ซึ่งหมายความว่าบิต 1 ซ้ายมือสุดของแต่ละ Octet คือบิต LSB ของแต่ละ Octet และบิต 0 ขวามือสุดคือบิต MSB ส่วน Start of Frame Delimiter หรือ SFD เป็นการบอกการเริ่มต้นของเฟรมข้อมูล ซึ่งจะถูกส่งออกไปก่อน Preamble และมีลำดับของบิตดังนี้คือ 10101011 เมื่อสัญญาณ TX_EN แอคทีพ Preamble และ SFD จะถูกส่งออกไปโดยมีลำดับดังแสดงในภาพประกอบ 2-19

Signal	Bit values of nibbles transmitted through MII																		
TXD0	X	1[a]	1	1	1	1	1	1	1	1	1	1	1	1	1	1[b]	1	D0[c]	D4[d]
TXD1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	D1	D5
TXD2	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	D2	D6
TXD3	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	D3	D7
TX_EN	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

[a]1st preamble nibble transmitted.
[b]1st SFD nibble transmitted.
[c]1st data nibble transmitted.
[d]D0 through D7 are the first eight bits of the data field from the Protocol Data Unit (PDU).

ภาพประกอบ 2-19 Preamble และ SFD ที่ถูกส่งผ่าน MII

(ที่มา : IEEE Std 802.3u-1995)

2.2.1.2.2.2 การรับข้อมูล SFD และข้อมูลที่ได้รับผ่าน MII จะแสดงดังภาพประกอบ 2-20

Signal	Bit values of nibbles received through MII																		
RXD0	X	1[a]	1	1	1	1	1	1	1	1	1	1	1	1	1	1[b]	1	D0[c]	D4[d]
RXD1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	D1	D5
RXD2	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	D2	D6
RXD3	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	D3	D7
RX_DV	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

[a]1st preamble nibble received.
[b]1st SFD nibble received.
[c]1st data nibble received.
[d]D0 through D7 are the first eight bits of the data field from the PDU.

ภาพประกอบ 2-20 ข้อมูลที่ได้รับซึ่งมี Preamble นำหน้า SFD

(ที่มา : IEEE Std 802.3u-1995)

2.2.1.2.3 Data ข้อมูลจะถูกรับส่งทีละ 4 บิต (Nibble)

2.2.1.2.4 End-of-Frame Delimiter (EFD) เมื่อสัญญาณ TX_EN หรือ RX_DV ไม่แอคทีพ EFD จะถูกส่งออกไปบนสัญญาณ TXD<3:0> หรือ RXD<3:0> ตามลำดับ

จากการศึกษาบทความและงานวิจัยที่เกี่ยวข้องกับเครื่องจัดการจราจรในระบบเครือข่ายในหัวข้อ 2.1 พบว่าวิธีการที่นำมาใช้เพื่อเพิ่มความเร็วในการทำงาน ได้แก่

1) การออกแบบอุปกรณ์ด้วย ASIC

2) การลดค่า Latency ที่เกิดขึ้นจากการทำงานของหน่วยความจำหรือจากการถ่ายโอนข้อมูลของหน่วยความจำ

3) การออกแบบโครงสร้างข้อมูลขึ้นมาโดยเฉพาะ

4) การใช้ซีพียูความเร็วสูง

5) การทำงานในลักษณะขนาน

สำหรับหัวข้อ 2.2.1 เป็นการศึกษาข้อมูลที่เกี่ยวข้องกับรูปแบบการอินเตอร์เฟสกับระบบเครือข่ายซึ่งจะถูกนำมาใช้ประกอบกับข้อพิจารณาในการออกแบบในบทที่ 3 ต่อไป