



ใบรับรองวิทยานิพนธ์  
บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)

ปริญญา

วิศวกรรมคอมพิวเตอร์

วิศวกรรมคอมพิวเตอร์

สาขา

ภาควิชา

เรื่อง การตัดแยกแพ็คเกจโดยการจัดกลุ่มพรีฟิกแบบสองมิติ

Two-Dimensional Packet Classification Based on Prefix Grouping

นามผู้วิจัย นายวรา ว่างฉาย

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

( รองศาสตราจารย์พันธุ์ปิติ เปี่ยมสง่า, D.Sc. )

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

( ผู้ช่วยศาสตราจารย์ไพรวรรณ วัฒนพงศ์, Ph.D. )

หัวหน้าภาควิชา

( รองศาสตราจารย์อนันต์ ผลเพิ่ม, Ph.D. )

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

( รองศาสตราจารย์กัญญา ชีระกุล, D.Agr. )

คณบดีบัณฑิตวิทยาลัย

วันที่ ..... เดือน ..... พ.ศ. ....

วิทยานิพนธ์

เรื่อง

การคัดแยกแพ็กเก็ตโดยการจัดกลุ่มพรีฟิกแบบสองมิติ

Two-Dimensional Packet Classification

Based on Prefix Grouping

โดย

นายวรา วิจารณ์

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)

พ.ศ. 2557

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

วรา ว่างฉาย 2557: การคัดแยกแพ็กเก็ตแบบ 2 มิติโดยการจัดกลุ่มพีริฟิก ปริญา  
วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์) สาขาวิศวกรรมคอมพิวเตอร์  
ภาควิชาวิศวกรรมคอมพิวเตอร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก:  
รองศาสตราจารย์พันธุ์ปิติ เปี่ยมสง่า, D.Sc. 50 หน้า

งานวิจัยนี้นำเสนอ โครงสร้างข้อมูลและขั้นตอนวิธีในการคัดแยกแพ็กเก็ต โดยการจัดกลุ่ม  
พีริฟิกแบบสองมิติของแต่ละแพ็กเก็ต เพื่อให้สามารถรองรับชุดตัวกรอง (Filter Set) ที่ปรับขนาด  
ได้ (Scalable) และสามารถนำไปใช้งานได้จริง รวมถึงรองรับการคัดแยกแพ็กเก็ตแบบไอพี ด้วย  
เทคนิคประมวลผลแบบขนานเพื่อตอบสนองต่อการร้องขอบริการที่แตกต่างกันได้อย่างรวดเร็ว  
งานวิจัยนี้นำเสนอผลการทดสอบทั้งในแง่ของการใช้หน่วยความจำของโครงสร้างข้อมูล และการ  
วัดจำนวนครั้ง que ที่อ่านข้อมูลจากหน่วยความจำเพื่อการคัดแยกแพ็กเก็ต โดยใช้ Benchmark ที่มีชื่อว่า  
ClassBench ซึ่งเป็นเครื่องมือในการสร้างชุดตัวกรองจำลอง โดยในการทดสอบนี้ได้เปรียบเทียบกับ  
งานวิจัยอื่น ๆ ซึ่งพบว่าโครงสร้างข้อมูลที่ใช้ชุดตัวกรองขนาด 50,000 ตัว ใช้หน่วยความจำ  
สูงสุดในกรณีที่เลวร้ายที่สุด ไม่เกิน 5 MBytes ซึ่งน้อยกว่างานวิจัยอื่น ๆ ประมาณ 1.75 เท่า และมี  
จำนวนครั้ง que ที่อ่านข้อมูลจากหน่วยความจำสูงสุด 23 ครั้งต่อแพ็กเก็ต ซึ่งเร็วกว่างานวิจัยอื่น ๆ  
ประมาณ 2 เท่า นอกจากนี้จำนวนของตัวกรองไม่มีนัยสำคัญต่อโครงสร้างข้อมูลและจำนวนครั้ง que  
ที่อ่านข้อมูลจากหน่วยความจำในการคัดแยกแพ็กเก็ต ทำให้งานวิจัยนี้มีความสามารถในการ  
ปรับขยายขนาดของชุดตัวกรองที่อาจเพิ่มขึ้นได้ในอนาคต

ลายมือชื่อนิติ

ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

Wara Wangchai 2014: Two-Dimensional Packet Classification Based on Prefix Grouping. Master of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Associate Professor Punpiti Piamsa-nga, D.Sc. 50 pages.

In this research, we propose a novel packet classification algorithm which is based on prefix grouping for 2 fields (2D packet classification). The algorithm takes advantage of grouping prefix so that it reduces the number of filters in filter set and it is scalable to build up data structure even the number of filter in filter set is much more higher. Besides, the research claims that the algorithm is suitable for parallel technique so that speed of classifying prefix is significantly increased.

The prefix grouping algorithm is compared to well-known existing algorithms by using benchmark tool named ClassBench. The experimental results show that our data structure takes the lowest memory requirement which is about 5 MBytes for the worst case. In addition, the number of memory access of our algorithm uses 23 times per packet which is 2 times as fast as other algorithms. In conclusion, the proposed algorithm improves the performance of packet classification especially scalable for large filter sets (50K), which it is the need today's solution of packet classification problem.

---

Student's signature

---

Thesis Advisor's signature

## กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. ทวีศักดิ์ กิจกาญจนารัตน์ และ  
รองศาสตราจารย์ ดร. พันธุ์ปิติ เปี่ยมสง่า อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ที่ให้คำปรึกษาใน  
การค้นคว้าวิจัยตลอดจนการตรวจแก้ไขวิทยานิพนธ์จนกระทั่งเสร็จสมบูรณ์ และกราบขอบพระคุณ  
คณาจารย์บัณฑิตวิทยาลัยทุกท่าน ที่ได้ให้ความกรุณาตรวจแก้ไขวิทยานิพนธ์ให้สมบูรณ์ยิ่งขึ้น

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่าน ที่ได้มอบความรู้  
อบรมสั่งสอนและคำแนะนำ อันเป็นประโยชน์อย่างยิ่งในการนำไปใช้ประโยชน์ทั้งในชีวิตด้านกร  
งานและชีวิตส่วนตัว

ด้วยประโยชน์อันใดเนื่องจากวิทยานิพนธ์เล่มนี้ ขอมอบแด่ทุกท่านที่มีใจรักการวิจัยและ  
พัฒนา โดยหวังว่าวิทยานิพนธ์นี้จะมีประโยชน์ต่อทุกท่านไม่มากนักน้อ

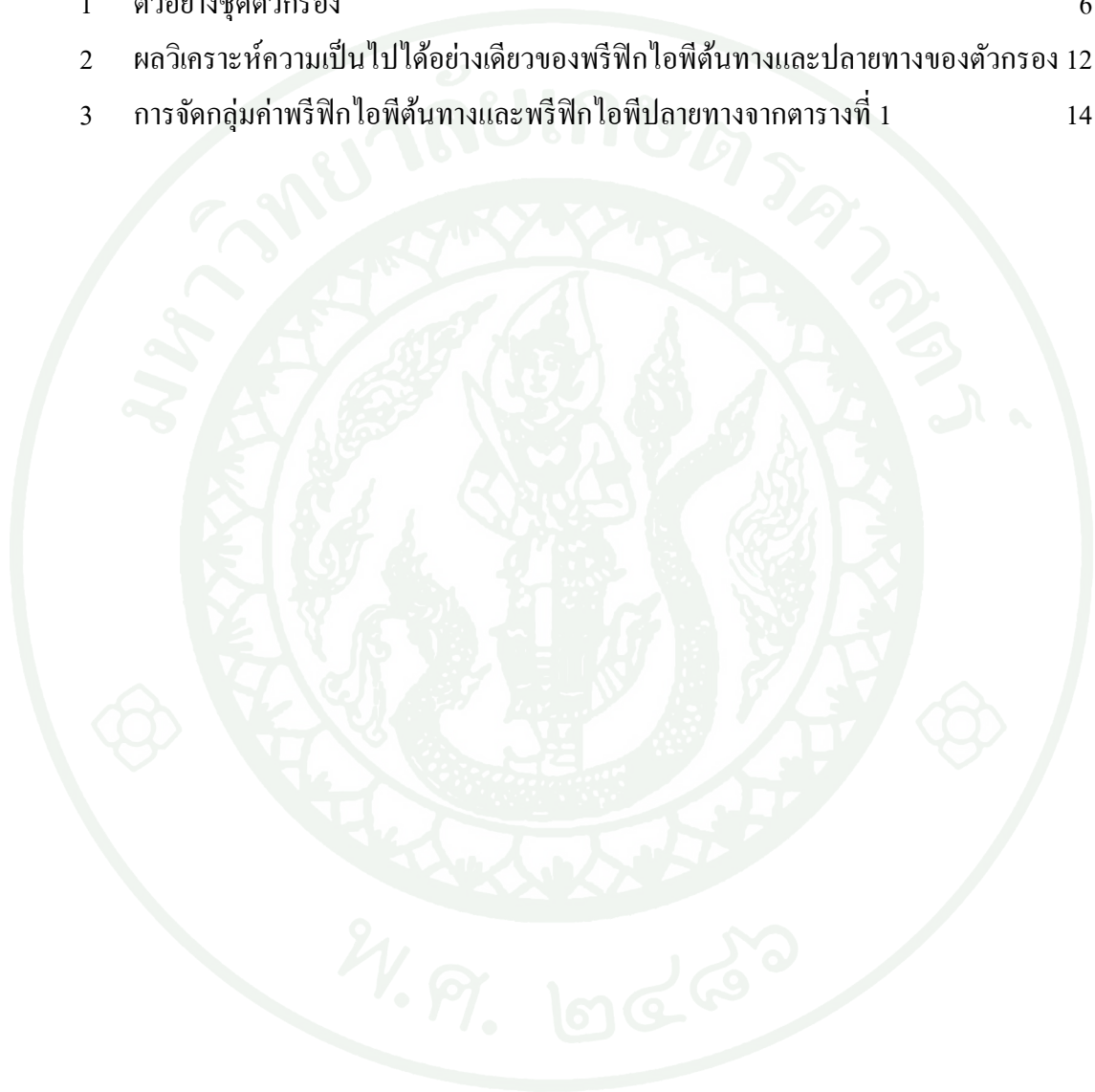
วรา วัจฉาย  
กรกฎาคม 2557

## สารบัญ

	หน้า
สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(3)
คำนำ	1
วัตถุประสงค์	4
การตรวจเอกสาร	8
อุปกรณ์และวิธีการ	12
ผลและวิจารณ์	19
สรุปและข้อเสนอแนะ	26
เอกสารและสิ่งอ้างอิง	28
ภาคผนวก	31
ภาคผนวก ก ผลการทดสอบหน่วยความจำที่ใช้ในโครงสร้างข้อมูลของตัวกรอง แต่ละชนิด	32
ภาคผนวก ข ผลการทดสอบจำนวนครั้งในการอ่านหน่วยความจำของตัวกรอง แต่ละชนิด	40
ภาคผนวก ค การจำลองชุดตัวกรองด้วย ClassBench	48
ประวัติการศึกษาและการทำงาน	50

## สารบัญตาราง

ตารางที่	หน้า
1 ตัวอย่างชุดตัวกรอง	6
2 ผลวิเคราะห์ความเป็นไปได้ของพรีฟิโอฟิตันทางและปลายทางของตัวกรอง	12
3 การจัดกลุ่มค่าพรีฟิโอฟิตันทางและพรีฟิโอฟิตันปลายทางจากตารางที่ 1	14



## สารบัญภาพ

ภาพที่		หน้า
1	โครงสร้างข้อมูลแบบ Cross-Product จากตารางที่ 3	14
2	โครงสร้างข้อมูล Cross-Product จากการปรับ โครงสร้างเป็นแถวลำดับของบิต	16
3	โครงสร้างข้อมูล Cross-Product ที่เก็บตัวกรองแบบความสำคัญสูงสุด	17
4	ตัวอย่างการคัดแยกไอพีต่อ โครงสร้างงานวิจัยที่นำเสนอ	18
5	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL	19
6	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW	20
7	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด IPC	20
8	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL โดยไม่รวม HSM	21
9	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด ACL	23
10	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด FW	23
11	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด IPC	24
12	เวลาในการสร้าง โครงสร้างข้อมูลในแต่ละชุดตัวกรอง (แสดงเป็นหน่วย นาฬิกา)	27
ภาพผนวกที่		
ก1	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL1	33
ก2	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL2	34
ก3	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL3	34
ก4	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL4	35
ก5	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL5	35
ก6	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW1	36
ก7	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW2	36
ก8	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW3	37
ก9	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW4	37

## สารบัญภาพ (ต่อ)

ภาพผนวกที่	หน้า	
ก10	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW5	38
ก11	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด IPC1	38
ก12	ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด IPC2	39
ข1	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด ACL1	41
ข2	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด ACL2	42
ข3	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด ACL3	42
ข4	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด ACL4	43
ข5	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด ACL5	43
ข6	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด FW1	44
ข7	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด FW2	44
ข8	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด FW3	45
ข9	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด FW4	45
ข10	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด FW5	46
ข11	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด IPC1	46
ข12	การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด IPC2	47
ค1	โครงสร้าง ClassBench	49

## การคัดแยกแพ็กเก็ตโดยการจัดกลุ่มพรีฟิกแบบสองมิติ

### Two-Dimensional Packet Classification Based on Prefix Grouping

#### คำนำ

ปัจจุบันความต้องการที่แตกต่างกันของผู้ใช้บริการและความสามารถในการจ่ายค่าบริการ อินเทอร์เน็ตของผู้ใช้มีความหลากหลาย เช่น ผู้ใช้บริการบางรายยินดีที่จะเสียค่าบริการจำนวนมาก เพื่อแลกกับการบริการที่มีคุณภาพ ได้แก่ โปรแกรมประยุกต์ประเภท Video Streaming ซึ่งต้องการแบนด์วิดท์ (Bandwidth) สูง ในขณะที่ผู้ให้บริการบางท่านอาจจะให้บริการพื้นฐานบนระบบ อินเทอร์เน็ตทั่วไปได้แก่ อีเมล (Email) ซึ่งการบริการแบบนี้เสียค่าใช้จ่ายน้อยกว่า ด้วยเหตุนี้ จึงทำให้เกิดการบริการอินเทอร์เน็ตที่มีความแตกต่างกัน เพื่อให้สามารถตอบสนองต่อความต้องการของผู้ใช้บริการได้ ผู้ให้บริการอินเทอร์เน็ต (Internet Service Provider (ISP)) จึงมีหน้าที่ตอบสนองต่อการร้องขอการบริการที่แตกต่างกันของผู้บริการ โดยที่การบริการต่าง ๆ นั้นจะถูกคัดแยกด้วยหลายปัจจัยตามความเหมาะสมของผู้ใช้บริการ เช่น การจำกัดแบนด์วิดท์, การกำหนดสิทธิ (Authorization) เพื่อกำหนดกลุ่มผู้บริการ, การคัดแยกเส้นทางการส่งต่อข้อมูลในเครือข่าย (Traffic Isolation) เป็นต้น ดังนั้นเพื่อให้การบริการในอินเทอร์เน็ตมีความแตกต่างกันอุปกรณ์จัดเส้นทาง (Router) จะต้องมีความสามารถในการคัดแยกข้อมูลหรือแพ็กเก็ต (Packet) ในเครือข่ายของผู้ใช้บริการได้ และสามารถกระทำต่อแพ็กเก็ตที่วิ่งเข้ามาในอุปกรณ์จัดเส้นทางได้ตามความเหมาะสม

โดยทั่วไป อินเทอร์เน็ตมีการให้บริการที่เรียบง่าย ที่เรียกว่า Best-Effort Service นั่นคือในการส่งต่อแพ็กเก็ตในแต่ละครั้งอุปกรณ์จัดเส้นทางจะส่งต่อแพ็กเก็ตหนึ่งไปยังอุปกรณ์จัดเส้นทางตัวถัดไปโดยอาศัย ไอพีปลายทาง (Destination IP Address, DA) ซึ่งอยู่ที่ส่วนหัวของแพ็กเก็ต (Packet Header) เพื่อเข้าสู่กระบวนการค้นหาเส้นทาง (Packet Forwarding) ว่าสำหรับ DA นี้จะต้องส่งต่อไปยังอุปกรณ์จัดเส้นทาง ตัวใดตัวใดนับจากอุปกรณ์จัดเส้นทางนั้น ๆ โดยพิจารณาจาก Routing Table ที่อยู่ของอุปกรณ์จัดเส้นทางแต่ละตัวของเครือข่าย อย่างไรก็ตามการให้บริการแบบนี้ไม่เหมาะสมต่อสถานการณ์ปัจจุบันที่ผู้บริการต้องการความแตกต่างกันในการให้บริการ เพราะทุก ๆ แพ็กเก็ตจะถูกกระทำอย่างเท่าเทียมกัน ไม่สนใจว่าข้อมูลต้นทางมาจากแหล่งใด ด้วยเหตุนี้ จึงได้มีการเปลี่ยนแปลงกระบวนการตรวจสอบแพ็กเก็ตแบบใหม่ เพื่อตอบสนองต่อผู้บริการที่มี

ความสำคัญแตกต่างกันได้ ซึ่งเรียกว่า การคัดแยกแพ็กเก็ต (Packet Classification) ซึ่งการบริการแบบนี้ไม่ได้อาศัยแค่ ไอพีปลายทาง ในการส่งต่อแพ็กเก็ตเท่านั้น แต่จะใช้ฟิลด์อื่น ๆ ที่อยู่ในข้อมูลส่วนหัวของแพ็กเก็ตรวมอยู่ด้วย ได้แก่ ไอพีต้นทาง, หมายเลขพอร์ตต้นทาง, หมายเลขพอร์ตปลายทาง, แพลกโปรโตคอล เป็นต้น โดยทั่วไปการคัดแยกแพ็กเก็ตจะมีความหลากหลายในการพิจารณาฟิลด์ที่ใช้ ซึ่งขึ้นอยู่กับเป้าหมายของการบริการ แต่ส่วนใหญ่จะใช้ห้าฟิลด์หรือเรียกว่าการคัดแยกแพ็กเก็ตแบบห้ามิติ

งานวิจัยนี้จะเสนอโครงสร้างข้อมูลและขั้นตอนวิธีในการคัดแยกแพ็กเก็ตแบบสองมิติ (Two-Dimensional Packet Classification) คือพิจารณาที่ ไอพีต้นทาง และ ไอพีปลายทาง แม้ว่าปัญหาการคัดแยกแพ็กเก็ตแบบสองมิตินั้นดูเหมือนว่าจะเป็นเพียงแค่ส่วนหนึ่งของการคัดแยกแพ็กเก็ตแบบห้ามิติ แต่ปัญหานี้ก็ยังมีมีความสำคัญที่สามารถประยุกต์ใช้งานได้หลายกรณี เช่น

1. โปรแกรมประยุกต์ประเภท VPN โดยเฉพาะ MPLS VPN ซึ่งต้องการค่า ไอพีต้นทาง และ ไอพีปลายทาง ในการสร้างระบบเครือข่าย VPN (Apipattanamongre and Kijkanjanarat, January – April 2004)

2. IP Multicast ซึ่งเป็นโปรโตคอลการส่งข้อมูลเฉพาะกลุ่มในเครือข่าย โดยจะใช้ไอพีต้นทางแหล่งเดียว และไอพีปลายทางหลายแห่ง ในการรับส่งข้อความแบบ Multicast ระหว่างกลุ่มเครือข่าย (Gupta and McKeown, 1999a) ซึ่งการส่งนี้จะกระทำเพียงครั้งเดียว ทำให้ประหยัด แบนด์วิดท์ในเครือข่าย IP Multicast ถูกนำไปใช้งานประเภทอินเทอร์เน็ตทีวีหรือ Streaming Media

3. IP Firewall ซึ่งเป็นโปรโตคอล (Protocol) เพื่อป้องกันการใช้วงจรรสื่อสารที่ไม่ได้รับอนุญาต โดยกำหนดค่าช่วงของไอพีต้นทาง และไอพีปลายทาง ต่อเครือข่ายนั้น ๆ เป็นต้น (Wencheng and Sahni, 2006)

งานวิจัยนี้จะเสนอโครงสร้างข้อมูลและขั้นตอนวิธีในการคัดแยกแพ็กเก็ตแบบสองมิติ โดยโครงสร้างข้อมูลอาศัยการจัดกลุ่มค่าพรีฟิก (Prefix) ในข้อมูลส่วนหัวคือ ไอพีต้นทางและไอพีปลายทาง และนำข้อมูลส่วนหัวทั้งสองค่ามา Cross-Product เพื่อให้ได้โครงสร้างข้อมูลที่ใช้หน่วยความจำน้อยที่สุด และสามารถรองรับเทคนิคแบบขนานในการคัดแยกไอพี จากการทดสอบทำให้ทราบว่าโครงสร้างข้อมูลที่เกิดจากการจัดกลุ่มข้อมูลส่วนหัวใช้หน่วยความจำเพียงแค่ 5 MBytes ในกรณีที่เลวร้ายที่สุด ต่อชุดตัวกรองขนาด 50,000 ตัว ซึ่งน้อยมาก และช่วยเพิ่มประสิทธิภาพในการคัดแยกแพ็กเก็ตโดยในการคัดแยกแพ็กเก็ตแต่ละครั้งมีจำนวนครั้งที่อ่านข้อมูล

จากหน่วยความจำระหว่าง 10 ถึง 20 ครั้งเท่านั้นโดยประมาณ ซึ่งมีความรวดเร็วมาก สามารถรองรับการบริการที่แตกต่างกันได้อย่างทันท่วงที ส่งผลให้ลดปัญหาคอขวดที่เกิดขึ้นต่ออุปกรณ์จัดเส้นทางได้อย่างดี



## วัตถุประสงค์

1. เพื่อลดเวลาในการส่งต่อแพ็กเก็ตโดยลดอัตราการเข้าถึงหน่วยความจำในแต่ละครั้งของการจับคู่แพ็กเก็ต
2. เพื่อให้โครงสร้างข้อมูลมีความต้องการพื้นที่หน่วยความจำที่น้อยเพื่อให้ประสิทธิภาพโดยรวมของการส่งต่อแพ็กเก็ตดีขึ้น
3. เพื่อลดเวลาในการเตรียมข้อมูลเพื่อสร้างโครงสร้างข้อมูล
4. เพื่อให้สามารถรองรับกับชุดตัวกรองที่มีขนาดใหญ่ขึ้นในอนาคตได้

## ปัญหาการคัดแยกแพ็กเก็ต

การคัดแยกแพ็กเก็ตเกิดเป็นปัญหาในระบบเครือข่ายที่นักวิจัยให้ความสำคัญเพื่อพัฒนาให้ระบบเครือข่ายมีประสิทธิภาพสูงสุดในการส่งต่อแพ็กเก็ต หัวข้อนี้จะกล่าวถึงหลักเกณฑ์ในการคัดแยกแพ็กเก็ตและในการวัดประสิทธิภาพของการคัดแยกแพ็กเก็ตดังนี้

### หลักในการคัดแยกแพ็กเก็ต

กำหนดให้มีชุดตัวกรองที่ประกอบด้วยชุดข้อมูลตัวกรองจำนวน  $n$  ตัวคือ  $F_1, F_2, F_3, \dots, F_n$  โดยที่ตัวกรอง  $F_i$  ใด ๆ ประกอบด้วยชุดข้อมูลที่มีค่าข้อมูลส่วนหัว  $K$  ค่า นั่นคือ  $H_{F_i}[1], H_{F_i}[2], H_{F_i}[3], \dots, H_{F_i}[k]$  อย่างเช่นใน IPv4 ชุดข้อมูลส่วนหัวประกอบด้วยค่า พรีฟิกของไอพีต้นทาง, พรีฟิกของไอพีปลายทาง, หมายเลขพอร์ตต้นทาง, หมายเลขพอร์ตปลายทาง และแฟล็กโปรโตคอล ซึ่งการ จับคู่ ค่าข้อมูลส่วนหัวจะแตกต่างกันตามรูปแบบข้อมูลในฟิลด์นั้น ๆ ซึ่งมีอยู่ด้วยกัน 3 ลักษณะคือ

1. การจับคู่แบบยาวที่สุด (Longest Matching) จะเป็นวิธีที่ใช้กับการจับคู่ค่าพรีฟิกต้นทางและปลายทาง โดยที่ค่าไอพีต้นทางของข้อมูลส่วนหัวของแพ็กเก็ตจะต้องจับคู่กับค่าพรีฟิกต้นทางที่มีพรีฟิกที่ยาวที่สุดในชุดตัวกรอง และค่าไอพีปลายทางของข้อมูลส่วนหัวของแพ็กเก็ตจะต้องจับคู่กับค่าพรีฟิกปลายทางที่มีพรีฟิกที่ยาวที่สุดในชุดตัวกรอง

2. การจับคู่แบบช่วง (Range Matching) จะเป็นวิธีที่ใช้กับการจับคู่หมายเลขพอร์ตต้นทางและปลายทาง โดย  $F_i$  คือตัวกรองใด ๆ ในชุดตัวกรองที่  $1 \leq i \leq n$  และ  $F_i$  ใด ๆ มีค่าหมายเลขพอร์ตต้นทางหรือหมายเลขพอร์ตปลายทางอยู่ในช่วง  $[\text{Min}_{F_i}, \text{Max}_{F_i}]$  โดยแพ็กเก็ต  $P$  ใด ๆ จะ จับคู่กับ  $F_i$  ถ้าค่าหมายเลขพอร์ตต้นทางและหมายเลขพอร์ตปลายทางของ  $P$  เป็นค่าหนึ่งในช่วง  $[\text{Min}_{F_i}, \text{Max}_{F_i}]$

3. การจับคู่โดยตรง (Exact Matching) จะเป็นวิธีที่ใช้กับการ จับคู่แฟล็กโปรโตคอลโดยที่แพ็กเก็ต  $P$  จะ จับคู่ กับ  $F_i$  ถ้าแฟล็กโปรโตคอลของ  $P$  มีค่าเท่ากับแฟล็กโปรโตคอลของ  $F_i$  ตัวอย่างของค่าแฟล็กโปรโตคอลเช่น TCP, UDP, ICMP เป็นต้น

สำหรับแพ็กเก็ต  $P$  ใด ๆ ที่  $P$  จะจับคู่ กับ  $F_i$  ก็ต่อเมื่อค่าข้อมูลส่วนหัวของ  $P$  จับคู่กับทุกค่าในข้อมูลส่วนหัวของ  $F_i$  ตามวิธีการจับคู่ ของแต่ละฟิลด์ และสุดท้ายแพ็กเก็ต  $P$  จะถูกกระทำโดยค่า

Action ที่ถูกกำหนดไว้ใน  $F_i$  เช่น Drop แเพ็กเก็ต, Forward แเพ็กเก็ตไปยังอุปกรณ์จัดเส้นทางตัวถัดไป เป็นต้น อย่างไรก็ตาม  $P$  หนึ่ง ๆ สามารถจับคู่กับ  $F_i$  ได้หลายค่า ขึ้นอยู่กับนโยบายของระบบเครือข่ายในการกำหนดค่าผลลัพธ์การจับคู่ โดยทั่วไปค่าผลลัพธ์สุดท้ายของการจับคู่จะอิงตามค่าความสำคัญของ  $F_i$  ที่ถูกกำหนดโดยระบบเครือข่ายนั้น ๆ

ตัวอย่างของการจับคู่ สมมติแพ็กเก็ต  $P$  วิ่งมาที่อุปกรณ์จัดเส้นทาง  $R$  ใด ๆ ที่มีชุดตัวกรองตามตารางที่ 1 โดยที่ \* หมายถึงค่า Don't Care และ  $P$  ประกอบด้วยข้อมูลส่วนหัว {ไอพีต้นทาง, ไอพีปลายทาง, หมายเลขพอร์ตต้นทาง, หมายเลขพอร์ตปลายทาง, โพรโทคอล} โดยแต่ละฟิลด์มีค่าเป็น {00101101, 01111010, 64, 1024, TCP} เมื่อมีการคัดแยกแพ็กเก็ต แพ็กเก็ต  $P$  จะจับคู่คือ 1) แบบยาวที่สุดจับคู่กับ  $F_5$  และ  $F_7$  เนื่องจาก  $P$  มีค่าฟิลด์ต้นทางและปลายทางที่ตกอยู่ในช่วง Subnet Mask ของ  $F_5$  และ  $F_7$ , 2) แบบช่วงจับคู่กับ  $F_2, F_4, F_5, F_6, F_7$  และ  $F_8$  เนื่องจากหมายเลขพอร์ตอยู่ในช่วงค่าพอร์ตต่ำสุดและสูงสุด ทั้ง 6 ตัวกรอง, 3) แบบจับคู่โดยตรง จับคู่กับแฟล็กโปรโตคอลคือ TCP และจากการรวมผลลัพธ์ด้วยวิธี AND จะทำให้ได้ตัวกรองสุดท้ายที่จับคู่กับ  $P$  คือ  $F_5$  และ  $F_7$  ถ้าอุปกรณ์จัดเส้นทาง  $R$  มีนโยบายของการคัดแยกแพ็กเก็ตโดยอิงตามค่าความสำคัญที่สูงที่สุดซึ่งกำหนดให้ตัวกรองเลขน้อยมีความสำคัญสูงกว่าตัวกรองเลขมาก จะได้ว่าตัวกรองที่จับคู่กับ  $P$  ที่มีความสำคัญสูงสุดคือ  $F_5$  และแพ็กเก็ต  $P$  จะถูกกระทำ Action เป็น  $Act_5$

งานวิจัยนี้เสนอขั้นตอนวิธีในการคัดแยกแพ็กเก็ตแบบสองมิติ ซึ่งวิธีการจับคู่ จะเป็นการจับคู่แบบยาวที่สุดต่อค่าข้อมูลส่วนหัวของฟิลด์ไอพีต้นทางและฟิลด์ไอพีปลายทาง

ตารางที่ 1 ตัวอย่างชุดตัวกรอง

#	ฟิลด์ต้นทาง	ฟิลด์ปลายทาง	พอร์ตต้นทาง	พอร์ตปลายทาง	โปรโตคอล	Action
F0	001101*	*	[64-64]	[0-65535]	UDP	$Act_0$
F1	10110*	1100*	[1024-1024]	[1024-1024]	TCP	$Act_1$
F2	10110*	1100*	[64-1222]	[1024-1024]	TCP	$Act_2$
F3	0100*	1*	[1024-1024]	[0-1024]	*	$Act_3$
F4	*	11001*	[0-65535]	[0-1024]	TCP	$Act_4$
F5	001*	0111*	[64-64]	[0-65535]	*	$Act_5$
F6	01011*	0111110*	[64-1222]	[1024-1024]	UDP	$Act_6$
F7	001*	0111*	[64-1222]	[0-65535]	*	$Act_7$

## ตารางที่ 1 (ต่อ)

#	พรีฟิกต้นทาง	พรีฟิกปลายทาง	พอร์ตต้นทาง	พอร์ตปลายทาง	โปรโตคอล	Action
F8	*	1100*	[64-1222]	[0-65535]	TCP	Act <sub>8</sub>
F9	10110*	1100*	[1048-4096]	[256-1024]	TCP	Act <sub>9</sub>
F10	001101*	*	[1024-1024]	[1024-1024]	ICMP	Act <sub>10</sub>

### การวัดประสิทธิภาพของการคัดแยกแพ็กเก็ต

โดยปกติการวัดประสิทธิภาพขั้นตอนวิธีจะคำนึงถึงปัจจัยที่สำคัญ คือ ความสามารถในการทำงานและความถูกต้องของผลลัพธ์ที่ได้จากขั้นตอนวิธีนั้น ๆ สำหรับหลักการคัดแยกแพ็กเก็ตจะมีเกณฑ์ที่สำคัญในการวัดประสิทธิภาพ คือขนาดหน่วยความจำที่ใช้สำหรับ โครงสร้างข้อมูลที่ใช้ แทนชุดตัวกรองและความรวดเร็วในการคัดแยกแพ็กเก็ตของขั้นตอนวิธีที่ใช้ในการคัดแยกแพ็กเก็ต

#### 1. ขนาดหน่วยความจำที่ใช้ในโครงสร้างข้อมูล (Memory Requirements)

งานวิจัยนี้จะวัดค่าพื้นที่หน่วยความจำของโครงสร้างข้อมูล ซึ่งเกณฑ์ที่ใช้ คือ ถ้าโครงสร้างข้อมูลใด ๆ ใช้พื้นที่หน่วยความจำน้อย แสดงว่าขั้นตอนวิธีนั้นมีประสิทธิภาพ เนื่องจากสามารถสร้างโครงสร้างข้อมูลนี้ลงในหน่วยความจำที่มีขนาดที่เทคโนโลยีปัจจุบันสามารถรองรับได้

#### 2. ความรวดเร็วในการคัดแยกแพ็กเก็ต (Packet Classification Rate)

จุดประสงค์ที่สำคัญของขั้นตอนวิธีคัดแยกแพ็กเก็ต คือ ความสามารถในการคัดแยกแพ็กเก็ตให้เร็วที่สุด เพื่อหลีกเลี่ยงปัญหาคอขวดที่เกิดขึ้นกับอุปกรณ์จัดเส้นทางในระบบเครือข่าย งานวิจัยนี้จะวัดความเร็วของการคัดแยกแพ็กเก็ตด้วยการวัดจำนวนครั้งที่เข้าถึงหน่วยความจำ (Number of Memory Accesses) ที่ค้นหาตัวกรองที่จับคู่ค่าพรีฟิกในชุดตัวกรองเมื่อคิดเทียบต่อ 1 แพ็กเก็ต โดยการวัดค่าได้กำหนดให้ Word Size ของหน่วยความจำมีขนาดเท่ากับ 16 Bytes เช่น ถ้าในการคัดแยกแพ็กเก็ต CPU จะต้องเข้าถึงหน่วยความจำทั้งหมด 320 Bytes ก็จะได้ค่าจำนวนครั้งที่เข้าถึงหน่วยความจำเท่ากับ 20 ในการวัดดังกล่าว ถ้าค่าจำนวนครั้งที่อ่านข้อมูลจากหน่วยความจำมีค่าน้อย ประสิทธิภาพในการคัดแยกแพ็กเก็ตก็จะสูงตามไปด้วย

## การตรวจเอกสาร

นับตั้งแต่อดีตจนถึงปัจจุบันได้มีงานทบทวนสำรวจงานวิจัยเกี่ยวกับการคัดแยกแพ็กเก็ตแบบสองมิติ (Dixit *et al.*, 2009; Gupta and McKeown, 2001; Taylor, 2005) ซึ่งงานวิจัยเหล่านี้มีความหลากหลายทั้งในแง่ของโครงสร้างข้อมูลและขั้นตอนวิธี โดยปกติสามารถคัดแยกขั้นตอนวิธีในการคัดแยกแพ็กเก็ตแบบสองมิติออกเป็นกลุ่ม ๆ ได้ดังนี้

### งานวิจัยทางด้านฮาร์ดแวร์

โดยทั่วไป การพัฒนาขั้นตอนวิธีในการคัดแยกแพ็กเก็ตสามารถคัดแยกออกเป็น 2 กลุ่มใหญ่ คือ การเน้นพัฒนาทางด้านซอฟต์แวร์และพัฒนาทางด้านฮาร์ดแวร์โดยใช้หน่วยความจำพิเศษที่เรียกว่า Ternary Content Addressable Memory (TCAMs) (Kempke, November, 1998; Montoye, June 1994; Woo, 2000) ซึ่งความสามารถของ TCAMs จะมีความรวดเร็วมากในการคัดแยกแพ็กเก็ตเพราะคุณสมบัติพื้นฐานที่สามารถกำหนดชุดตัวกรองลงไปได้โดยตรงในหน่วยความจำ นั่นคือกระบวนการค้นหาจะทำได้ใน  $O(1)$  อย่างไรก็ตาม TCAMs ก็มีข้อด้อยที่ไม่เหมาะสมต่อการนำมาใช้งานคัดแยกแพ็กเก็ตดังนี้

1. TCAMs เป็นฮาร์ดแวร์ที่ใช้พลังงานจำนวนมากเมื่อเทียบกับหน่วยความจำทั่ว ๆ ไป เพราะข้อมูลหนึ่ง ๆ ที่บันทึกใน TCAMs จะต้องใช้ทรานซิสเตอร์จำนวนมาก
2. เนื่องจาก TCAMs ใช้ Transistor จำนวนมากทำให้มีราคาสูง
3. TCAMs ไม่สามารถนำข้อมูลประเภทช่วงเช่นค่าหมายเลขพอร์ตมาใส่ในหน่วยความจำได้โดยตรง โดยจะต้องแปลงค่าช่วงเป็นค่าฟริฟิกแล้วจึงนำมาใส่ในหน่วยความจำ ซึ่งเป็นวิธีที่มีความซับซ้อน และสิ้นเปลืองพื้นที่หน่วยความจำ

จากข้อจำกัดดังกล่าวทั้งหมดของ TCAMs ทำให้ TCAMs ยังไม่เป็นที่นิยมที่จะนำมาใช้งานอย่างจริงจัง ด้วยเหตุผลนี้เองที่ทำให้การเน้นพัฒนาทางด้านซอฟต์แวร์ยังคงมีความสำคัญ

## งานวิจัยทางด้านซอฟต์แวร์

กลุ่มงานวิจัยทางด้านซอฟต์แวร์คือการพัฒนาขั้นตอนวิธีภายใต้เทคโนโลยีหน่วยความจำพื้นฐานจำพวก Static Random-Access Memory (SRAM) ซึ่งสามารถคัดแยกเป็นกลุ่มย่อย ๆ ตามโครงสร้างข้อมูลและขั้นตอนวิธีดังนี้

### 1. ขั้นตอนวิธีแบบเชิงเส้น (Linear Search Algorithm)

งานวิจัยรูปแบบนี้ (Gupta and McKeown, 1999b; Woo, 2000) จะมีโครงสร้างข้อมูลที่ไม่ซับซ้อน นั่นคือแถวลำดับ (Array) ของ Index ที่บอกถึงกลุ่มของตัวกรองที่ค่าฟิลด์อยู่ในช่วงขอบเขตเดียวกัน โดยที่แต่ละช่องของแถวลำดับจะมีตัวชี้ไปยังโครงสร้างข้อมูล Tree ที่จัดเก็บตัวกรองที่ตกอยู่ในกลุ่มนี้ สำหรับกระบวนการคัดแยกแพ็กเก็ตจะเป็นแบบเชิงเส้น ในแถวลำดับและเป็นแบบไบนารี ใน Tree ซึ่งจะเห็นได้ว่ากระบวนการคัดแยกแพ็กเก็ตที่ถึงแม้จะมีกระบวนการแบบไบนารี ซึ่งมีความซับซ้อนเป็น  $O(\log n)$  มาใช้ในการคัดแยกแพ็กเก็ตใน Tree แต่กระบวนการคัดแยกแพ็กเก็ตโดยรวมจะมีความช้ามาก อันเนื่องมาจากกระบวนการเชิงเส้น ซึ่งมีความซับซ้อนเป็น  $O(n)$  ซึ่งไม่เหมาะสมต่อการนำมาใช้งานจริง ๆ

### 2. ขั้นตอนวิธีแบบ Trie (Trie-Based Algorithm)

รูปแบบขั้นตอนวิธีแบบ Trie (Baboescu *et al.*, 2003; Baboescu and Varghese, 2005; Srinivasan *et al.*, 1998) จะเหมือนกับ Binary-Tree โดยที่แต่ละโหนดจะมีตัวชี้สองค่า คือ 0 หรือ 1 ซึ่งวิธีการสร้างโครงสร้าง Trie จะใช้ค่าเลขที่อยู่ไอพีทั้ง 32 บิต นำมาสร้างเป็นโหนดใน Trie สำหรับกระบวนการคัดแยกแพ็กเก็ตจะกระทำภายใต้ Trie โดยจะค้นหาทีละบิตจากโหนด Root จนถึงค่าพรีฟิกสุดท้ายที่โหนด Leaf (ใช้การจับคู่แบบยาวที่สุด) ข้อดีของโครงสร้างข้อมูลนี้ คือ ไม่ซับซ้อนโดยเฉพาะเมื่อใช้ชุดตัวกรองที่มีขนาดเล็ก แต่เมื่อชุดตัวกรอง มีขนาดใหญ่มา ๆ ขนาดของ Trie ก็จะถูกกว้าง (Breadth) และสูง (Height) ตามไปด้วย ส่งผลกระบวนการคัดแยกแพ็กเก็ตภายใต้โครงสร้างข้อมูลนี้ช้า อีกทั้งยังใช้หน่วยความจำมากขึ้นแบบ Exponential ซึ่งไม่เหมาะสมในการนำมาใช้งานจริงที่ชุดตัวกรองมีขนาดใหญ่มาได้

### 3. ขั้นตอนวิธีแบบ Heuristic (Heuristic-Based Algorithm)

งานวิจัยในกลุ่มนี้ได้แก่ Recursive Flow Classification (RFC) (Gupta and McKeown, 1999a) ซึ่งโครงสร้างข้อมูลเกิดจากการแบ่งค่าฟิลด์ออกเป็นกลุ่ม ๆ ซึ่งแต่ละกลุ่มจะมีโครงสร้างเป็นแถวลำดับที่มีขนาดเป็น  $2^{(bit)}$  โดยที่ค่า *bit* คือค่าจำนวนบิตที่ใช้ในการแบ่งฟิลด์ เช่น ในโครงสร้างข้อมูลของแพ็กเก็ตแบบสองมิติ คือ ฟรีฟีกไอพีต้นทางและฟรีฟีกไอพีปลายทาง จะถูกจัดกลุ่มโดยการแบ่งออกเป็นกลุ่มละ 16 บิต ซึ่งจะได้ 4 กลุ่ม เป็นต้น และแต่ละกลุ่มก็จะถูกจับคู่ซึ่งกันและกัน ก่อให้เกิดโครงสร้างข้อมูลใหม่เป็นตารางข้อมูล และจะกระทำเช่นนี้เรื่อย ๆ ในทุกเฟส (Phase) จนสุดท้ายจะได้ตารางข้อมูลที่เกิดจากการจับคู่ของตารางข้อมูลก่อนหน้าทั้งหมด สำหรับกระบวนการคัดแยกแพ็กเก็ตจะกระทำภายใต้ชุดข้อมูลที่ถูกแบ่งตามแต่ละเฟสของการแบ่งข้อมูล การค้นหาแพ็กเก็ตจะกระทำในทุก ๆ เฟสและในแต่ละเฟสจะจับคู่ข้อมูลอย่างพร้อม ๆ กันในทุก ๆ ฟิลด์ (Parallel) นั่นคือยิ่งค่าเฟสที่แบ่งมีค่าน้อยเท่าใด การค้นหาจะมีความเร็วมากเท่านั้น ซึ่ง RFC จัดเป็นขั้นตอนวิธีหนึ่งที่มีความเร็วมากที่สุดในการคัดแยกแพ็กเก็ต เพราะในกระบวนการค้นหาจะเป็นการจับคู่ Index ของตารางข้อมูลโดยตรง อย่างไรก็ตาม ด้วยโครงสร้างข้อมูลที่ซับซ้อนและในชุดของกลุ่มข้อมูลที่ถูกรวบรวมมีการใช้หน่วยความจำเป็นจำนวนมากอันเนื่องมาจากแถวลำดับที่ใช้เก็บข้อมูลมีขนาดใหญ่ ทำให้ RFC ไม่สามารถรองรับชุดตัวกรองที่มีขนาดใหญ่ได้ อย่างไรก็ตามได้มีการวิจัย Hierarchical Space Mapping (HSM) (Xu *et al.*, 2005) ซึ่งแก้ไขข้อเสียของ RFC ที่จะต้องใช้หน่วยความจำจำนวนมากในการสร้างโครงสร้างข้อมูล โดยวิธีการ คือ แทนที่จะต้องแบ่งค่าฟิลด์ออกเป็นกลุ่ม ๆ HSM จะสร้างโครงสร้างแถวลำดับตามชนิดของฟิลด์ที่แตกต่างกัน เช่น โครงสร้างข้อมูลของแพ็กเก็ตแบบสองมิติ จะมีโครงสร้างแถวลำดับในเฟสแรกแค่ 2 โครงสร้างเท่านั้นคือ ฟรีฟีกไอพีต้นทางและฟรีฟีกไอพีปลายทาง (จะได้ขนาดแถวลำดับเป็น  $2^{32}$ ) และเช่นเดียวกับ RFC ที่เฟสถัดไปจะเป็นการสร้างตารางข้อมูลที่เกิดจากการจับคู่ตารางก่อนหน้าจนได้ตารางสุดท้ายที่เกิดจากการจับคู่ทั้งหมด สำหรับกระบวนการคัดแยกแพ็กเก็ตจะต่างกับ RFC ที่ไม่ได้ใช้วิธีจับคู่ Index กับตารางข้อมูลโดยตรง แต่จะใช้วิธีค้นหาแบบไบนารีที่ตารางข้อมูลเพื่อหาค่า Index ของตารางข้อมูลในเฟสถัดไป ซึ่งวิธีนี้เป็นผลเสียต่อ HSM เนื่องจากกระบวนการแบบไบนารีมีความซับซ้อนเป็น  $O(\log n)$  ในขณะที่การจับคู่ Index ของ RFC มีความซับซ้อนเป็น  $O(1)$

งานวิจัย HyperCuts (Singh *et al.*, 2003) และ HyperSplit (Qi *et al.*, 2009) เป็นงานวิจัยที่มีความซับซ้อนเช่นเดียวกับ RFC โดยแนวคิดของงานวิจัยนี้คือการกำหนดให้ฟิลด์เป็นมิติ (Dimension) ซึ่งโดยปกติแต่ละมิติจะมีความเหลื่อมล้ำของชุดข้อมูลนั้น ๆ อยู่ (Field Overlap) ซึ่งผู้วิจัยจะใช้เวลาเหลื่อมล้ำนั้นนำมาสร้างเป็นโครงสร้างข้อมูลแบบ Multiway-Tree โดยการหาความยาวฟรีฟีกของตัวกรองในแต่ละมิติเพื่อกำหนดจำนวนโหนด (Way) ที่เป็นไปได้ในระดับ

ถัดไปของ Tree อย่างไรก็ตามที่ HyperCuts จะแบ่งจำนวน โหนดในแต่ละมิติเท่า ๆ กันเป็นจำนวนเท่าของ 2 และในแต่ละโหนดของ Tree สามารถมีได้มากกว่า 1 มิติ แต่ HyperSplit จะแบ่งจำนวน โหนดตามค่าขอบของตัวกรองและการแบ่งแต่ละครั้งจะกระทำแค่มิติเดียวเท่านั้น สำหรับการตัดแยกแพ็คเกจจะกระทำภายใต้ Tree นี้ โดยการจับคู่แพ็คเกจเกิดตามค่าฟิลด์ในแต่ละโหนด จนสุดท้ายการเดินทางภายใน Tree (Packet Traversal) สิ้นสุดที่โหนด Leaf งานวิจัยนี้มีข้อเสียคือค่าความสูงของ Tree (Tree Height) มีค่ามากทำให้กระบวนการค้นหาช้า อีกทั้งกระบวนการสร้างโครงสร้างข้อมูลจะต้องมีการจองหน่วยความจำเป็นจำนวนมาก ทำให้ไม่สามารถรองรับชุดตัวกรองที่มีขนาดใหญ่ได้



## อุปกรณ์และวิธีการ

จากงานวิจัยทั้งหมดที่กล่าวมาข้างต้น แม้จะมีความสามารถในการคัดแยกแพ็กเก็ต แต่กระนั้นก็ยังมียกข้อจำกัด นั่นคือความสามารถในการรองรับชุดตัวกรองที่มีขนาดใหญ่ได้ (Scalable) ซึ่งปัจจัยนี้เป็นสิ่งสำคัญในการพัฒนาขั้นตอนวิธีในการคัดแยกแพ็กเก็ต งานวิจัยนี้เสนอขั้นตอนวิธีในการคัดแยกแพ็กเก็ตแบบสองมิติโดยอาศัยการจัดกลุ่มข้อมูลส่วนหัวเพื่อจำกัดการใช้หน่วยความจำในการสร้างโครงสร้างข้อมูล ซึ่งจากการวิจัยทำให้ทราบว่าขั้นตอนวิธีที่นำเสนอนี้สามารถรองรับชุดข้อมูลชุดตัวกรองที่มีขนาดใหญ่ได้ และด้วยกระบวนการ Cross-Product ภายใต้อการจัดกลุ่มข้อมูล ทำให้ขั้นตอนวิธีนี้สามารถค้นหาข้อมูลได้อย่างรวดเร็วอีกด้วย และรวมถึงโครงสร้างข้อมูลที่ไม่ซับซ้อน ทำให้ขั้นตอนวิธีนี้สามารถนำมาติดตั้งลงในอุปกรณ์ฮาร์ดแวร์ได้อย่างง่าย ซึ่งแสดงให้เห็นว่าขั้นตอนวิธีนี้สามารถนำมาประยุกต์ใช้งานได้จริงด้วยเทคโนโลยีปัจจุบัน

แนวคิดของวิธีงานวิจัยนี้ได้แรงบันดาลใจมาจากการสังเกตผลการศึกษาของงานวิจัย DCFL (Taylor and Turner, 2005a) โดย David E. Taylor และ Jonathan S. Turner ซึ่งงานวิจัย DCFL ได้ให้ข้อสังเกตว่ารูปแบบชุดตัวกรองใด ๆ ในปัจจุบัน จะมีค่าซ้ำ ๆ กันมากตามแต่ละค่าฟิลด์ นั่นคือถ้าจัดกลุ่มค่าฟิลด์ด้วยกัน หรือจัดกลุ่มค่าหมายเลขพอร์ตหรือจัดกลุ่มค่าแฟล็กโปรโตคอล จำนวนกลุ่มที่ได้จะมีค่าน้อยมากเมื่อเทียบกับจำนวนตัวกรองที่อยู่ในชุดตัวกรอง แสดงให้เห็นว่าแท้จริงแล้วจำนวนตัวกรองที่เป็นไปได้ทั้งหมดมีค่าความเป็นไปได้เพียงอย่างเดียว (Uniqueness) น้อยมาก จึงเกิดแนวคิดให้กับผู้วิจัยว่า น่าจะจัดตัวกรองออกเป็นกลุ่ม ๆ ตามค่าข้อมูลส่วนหัวและให้มีการคัดแยกแพ็กเก็ตแยกตามกลุ่มแทน งานวิจัยนี้ได้วิเคราะห์และสรุปจำนวนตัวกรองดังตารางที่ 2 โดยตัวกรองที่ใช้นำมาจากงานวิจัย ClassBench (Taylor and Turner, 2005b) ซึ่งแบ่งตัวกรองออกเป็น 3 ชุดหลักคือ Access Control List (ACL), Firewall (FW) และ IP Chain (IPC)

**ตารางที่ 2** ผลวิเคราะห์ค่าความเป็นไปได้ของไอพีต้นทางและไอพีปลายทางของตัวกรองในชุดตัวกรอง

ชุดตัวกรอง	จำนวน	พีรีฟิคต้นทาง	พีรีฟิคปลายทาง	จำนวนพีรีฟิคสูงสุดในการจับคู่แต่ละแพ็กเก็ต
ACL1	748	280	109	2
ACL2	627	368	182	4

## ตารางที่ 2 (ต่อ)

ชุดตัวกรอง	จำนวน	พรีฟิกล้นทาง	พรีฟิกลบทาง	จำนวนพรีฟิกสูงสุด
				ในการจับคู่แต่ละแพ็กเก็ต
ACL3	2410	546	480	3
ACL4	3074	1244	789	3
ACL5	4562	1179	306	2
FW1	286	87	43	3
FW2	69	27	41	1
FW3	193	21	22	2
FW4	270	113	38	3
FW5	171	38	34	2
IPC1	1724	637	380	4
IPC2	192	55	56	2

จากตารางที่ 2 จะเห็นได้ว่าจำนวนตัวกรองจริง ๆ ในแต่ละชุดจะมีค่าความเป็นไปได้ อย่างเดียว น้อยมาก เช่น ACL5 มีตัวกรองจำนวน 4,562 ตัว แต่เมื่อจัดกลุ่มตามค่าไอพีปลายทาง จะเหลือ เพียงแค่ 306 ตัว หรือลดลงประมาณ 14 เท่า นอกจากนี้จำนวนพรีฟิกโดยเฉลี่ยที่จับคู่กับแพ็กเก็ต หนึ่ง ๆ มีจำนวนพรีฟิกค่าตั้งแต่ 2 จาก 4,562 ตัวเท่านั้น ซึ่งผลที่ได้เป็นไปตามสมมติฐานที่ผู้วิจัยตั้งไว้

จากการค้นพบนี้ทำให้ผู้วิจัยได้พัฒนาโครงสร้างข้อมูลและขั้นตอนวิธีการคัดแยกแพ็กเก็ตแบบสองมิติ โดยการสร้างโครงสร้างข้อมูลด้วยการจัดกลุ่มค่าข้อมูลส่วนหัวแทนการนำค่าตัวกรองทั้งหมดในชุดตัวกรองมาใช้ ซึ่งทำให้ลดจำนวนการใช้พื้นที่หน่วยความจำที่ไม่จำเป็น และส่งผลถึงความสามารถในการคัดแยกแพ็กเก็ตที่รวดเร็วขึ้นเนื่องจากโครงสร้างข้อมูลที่มีขนาดเล็ก สำหรับรายละเอียดของขั้นตอนวิธีจะถูกแยกออกเป็น 3 ส่วนคือ 1) กระบวนการจัดกลุ่มชุดตัวกรองก่อนที่จะนำมาใช้ในโครงสร้างข้อมูล 2) รายละเอียดโครงสร้างข้อมูล 3) กระบวนการคัดแยกแพ็กเก็ตตามค่าไอพีต้นทางและไอพีปลายทางของแพ็กเก็ต

### กระบวนการจัดกลุ่มชุดตัวกรอง

จากผลของการจัดกลุ่มของตัวกรองต่าง ๆ ของตารางที่ 1 ได้ผลดังแสดงในตารางที่ 3 ซึ่งสามารถนำค่าตัวกรองเหล่านี้มาจัดลงใน โครงสร้างข้อมูลแบบ Cross-Product ดังแสดงในภาพที่ 1

ตารางที่ 3 การจัดกลุ่มค่าพรีฟิกไอพีต้นทางและพรีฟิกไอพีปลายทาง จากตารางที่ 1

พรีฟิก	เลขที่อยู่ต้นทาง	พรีฟิก	เลขที่อยู่ปลายทาง
	ตัวกรอง		ตัวกรอง
*	F4, F8	*	F0, F10
001*	F5, F7	1*	F3
0100*	F3	0111*	F5, F7
01011*	F6	1100*	F1, F2, F8, F9
10110*	F1, F2, F9	11001*	F4
001101*	F0, F10	011110*	F6

#### Group of destination prefixes

	*	1*	0111*	1100*	11001*	011110*
Group of source prefixes	*			F8	F4	
001*			F5,F7			
0100*		F3				
01011*						F6
10110*				F1,F2,F9		
001101*	F0,F10					

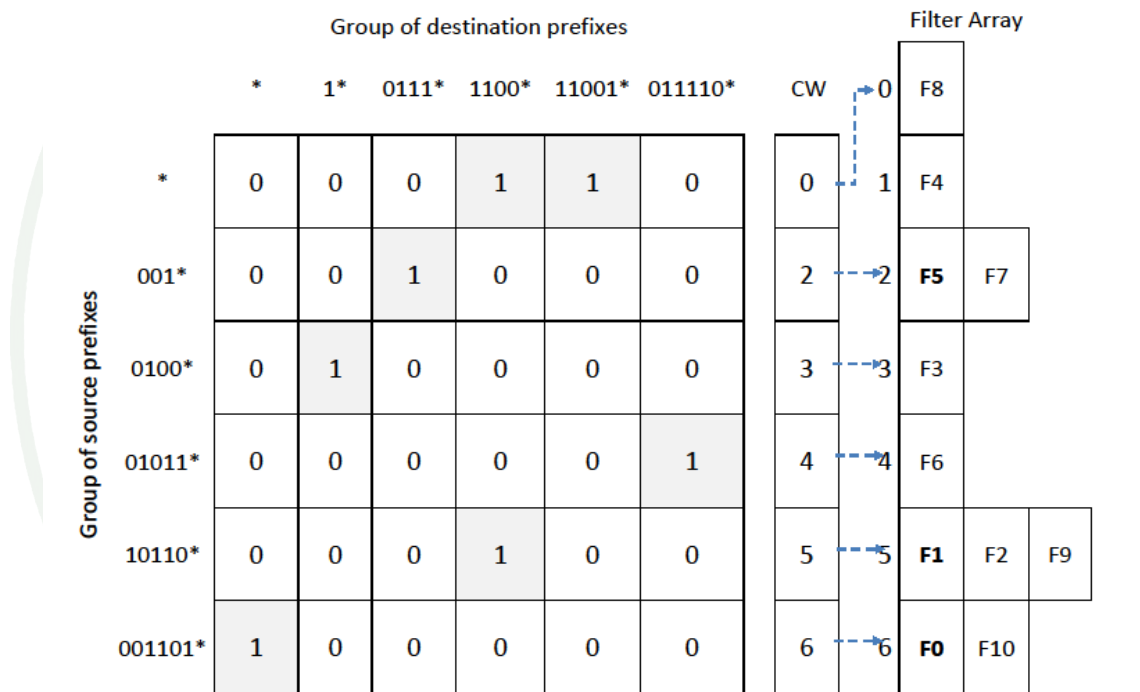
ภาพที่ 1 โครงสร้างข้อมูลแบบ Cross-Product จากตารางที่ 3

## โครงสร้างข้อมูล

ในโครงสร้างข้อมูลจะใช้กลุ่มตัวกรองที่ได้มีการทำ Pre-Computation แล้ว โดยโครงสร้างข้อมูลอย่างง่ายประกอบด้วย Cross-Product ของค่าฟิลด์พรีฟิกต้นทางและพรีฟิกปลายทาง และค่าภายใน Cross-Product ซึ่งจะเก็บหมายเลขตัวกรอง (Filter Number) ที่ใช้อ้างถึงตัวกรองที่สอดคล้องกับค่าพรีฟิกดังแสดงภาพที่ 1 กระบวนการค้นหาตัวกรองจะเกิดขึ้นจากการจับคู่ค่าพรีฟิกทั้งพรีฟิกต้นทาง (ในด้านแนวนอน) และพรีฟิกปลายทาง (ในด้านแนวตั้ง) พร้อม ๆ กัน โดยการจับคู่ค่าพรีฟิกสามารถใช้ขั้นตอนวิธีที่มีประสิทธิภาพเช่น Helix (Rojas-Cessa *et al.*, 2010) หรือขั้นตอนวิธีอื่น ๆ ในการจับคู่ค่าพรีฟิก (Apipattanamongre and Kijkanjanarat, January – April 2004) และเมื่อทราบค่า Cross-Product จากการจับคู่ค่าฟิลด์ทั้งสองแล้ว จะทราบค่าตัวกรองที่จับคู่จากตำแหน่งของ Cross-Product ซึ่งถ้าในการ จับคู่ ให้ผลลัพธ์มากกว่า 1 ตัวก็จะอิงตามนโยบายของอุปกรณ์จัดเส้นทางเพื่อหาค่าผลลัพธ์ที่แท้จริง

แม้โครงสร้างข้อมูลจากภาพที่ 1 จะเป็นโครงสร้างข้อมูลที่ไม่ซับซ้อนและมีความรวดเร็วในการคัดแยกแพ็กเก็ต แต่โครงสร้างข้อมูลนี้มีข้อเสียที่สำคัญ คือ ในทางปฏิบัติเมื่อชุดตัวกรองมีขนาดใหญ่มาก เช่น FW เมื่อจัดกลุ่มค่าพรีฟิกต้นทางและพรีฟิกปลายทางพบว่าจำนวนตัวกรองที่ Cross-Product กันในแต่ละช่องในตารางมีเป็นจำนวนมาก ส่งผลให้ช่องหนึ่ง ๆ ใช้หน่วยความจำค่อนข้างสูงเพื่อรองรับกับจำนวนตัวกรองที่มาก ซึ่งจะทำให้ขนาดโครงสร้างข้อมูลที่ใช้ในการคัดแยกแพ็กเก็ตมากขึ้นตามไปด้วย ดังนั้นจึงได้จัดโครงสร้างใหม่โดยแยกค่าตัวกรองออกเป็นแถวลำดับต่างหาก และให้ค่า Cross-Product เป็นแถวลำดับสองมิติของบิตโดยถ้าช่องใดมีค่าบิตเป็น 1 หมายถึงช่องนั้นมีตัวกรองอยู่ แต่ถ้าค่าบิตเป็น 0 แสดงว่าไม่มีตัวกรองอยู่ในช่องนั้น โดยจะมี Code Word (CW) เป็นค่า Index ที่ใช้อ้างอิงในแถวลำดับของตัวกรองเพื่อให้ทราบค่า “ตำแหน่งเริ่มต้น” ในแถวลำดับของตัวกรองที่สอดคล้องกับค่า Cross-Product ในแต่ละแถว เช่น จากภาพที่ 2 ค่า CW ของแถวแรกมีค่าเป็น 0 ซึ่งอ้างถึงตำแหน่งช่องตำแหน่งแรกที่มีค่าบิตเป็น 1 หรือตำแหน่งค่า Cross-Product (\*, 1100\*) ซึ่ง CW นี้จะใช้อ้างตำแหน่ง Index 0 ของแถวลำดับของตัวกรอง นั่นคือค่า F8 อย่งไรก็ดีในแถวแรกนี้มีค่าบิตเป็น 1 ที่ตำแหน่ง (\*, 11001\*) และเป็นตำแหน่งที่ 2 ถัดจากตำแหน่งแรกคือ (\*, 1100\*) ซึ่งในการอ้าง Index ในแถวลำดับของตัวกรองจะต้องใช้ค่า CW ของแถวนี้ คือค่า 0 บวกกับค่าตำแหน่งที่มีค่าบิตเป็น 1 นับจากตำแหน่งแรก นั่นคือค่า 1 ดังนั้นค่า Index ที่ใช้บอกตำแหน่งแถวลำดับของตัวกรองของ (\*, 11001\*) คือ  $0 + 1 = 1$  นั่นคือ Index 1 และได้ค่าตัวกรองเป็น F4

ในทางปฏิบัติ การจับคู่ค่าตัวกรองจะสนใจตัวกรองที่จับคู่กับแพ็กเก็ตหนึ่ง ๆ ได้ดีที่สุดเพียงแค่ค่าเดียว โดยอุปกรณ์จัดเส้นทางจะกระทำ Action ที่เป็นไปตามที่ระบุในตัวกรองนั้น ในงานวิจัยนี้ ผู้วิจัยจะใช้นโยบายการคัดแยกแพ็กเก็ตที่อิงตามค่าความสำคัญสูงสุดของตัวกรอง โดยกำหนดให้ตัวกรองที่มีเลขน้อยมีค่าความสำคัญสูงและตัวกรองที่มีเลขมากให้มีค่าความสำคัญต่ำ ตัวกรองที่จับคู่กับแพ็กเก็ตได้ดีที่สุด คือตัวกรองที่มีเลขน้อยที่สุดในบรรดาตัวกรองที่ถูกจับคู่ทั้งหมด และจากการกำหนดนโยบายนี้ โครงสร้างข้อมูลในภาพที่ 2 สามารถเปลี่ยนเพื่อให้สอดคล้องกับนโยบายโดยปรับแถวลำดับของตัวกรองให้เก็บเฉพาะตัวกรองที่มีค่าความสำคัญสูงที่สุดเท่านั้น ดังภาพที่ 3 ซึ่งจากการเปลี่ยนแปลงนี้ส่งผลให้ขนาดของหน่วยความจำที่ใช้ในโครงสร้างข้อมูลลดลงด้วย



ภาพที่ 2 โครงสร้างข้อมูล Cross-Product จากการปรับโครงสร้างเป็นแถวลำดับของบิต

Group of source prefixes	Group of destination prefixes						CW	Filter array (Highest priority)	
	*	1*	0111*	1100*	11001*	011110*		1	F8
*	0	0	0	1	1	0	0	2	F4
001*	0	0	1	0	0	0	2	3	F5
0100*	0	1	0	0	0	0	3	4	F3
01011*	0	0	0	0	0	1	4	5	F6
10110*	0	0	0	1	0	0	5	6	F1
001101*	1	0	0	0	0	0	6	7	F0

ภาพที่ 3 โครงสร้างข้อมูล Cross-Product ที่เก็บตัวกรองแบบนโยบายความสำคัญสูงสุด

### ขั้นตอนวิธีในการคัดแยกแพ็กเก็ต

เทคโนโลยีในปัจจุบันมีความก้าวหน้ามาก สามารถประมวลผลงานต่าง ๆ ภายในระยะเวลาอันสั้นโดยใช้วิธี Parallel Computing กระบวนการนี้อธิบายอย่างง่ายคือการประมวลผลข้อมูลโดยการแบ่งงานให้ CPU แต่ละตัวประมวลผลพร้อม ๆ กัน (Multi-Processing) และนำผลลัพธ์จากการประมวลผลของ CPU แต่ละตัวมารวมกันเพื่อจบการทำงาน ซึ่งการประมวลผลดังกล่าวมีความสำคัญต่องานที่มีชุดข้อมูลขนาดใหญ่ เพราะจะได้ผลลัพธ์ที่รวดเร็วกว่าการประมวลผลโดยใช้วิธี Sequential Computing งานวิจัยนี้ผู้วิจัยกำหนดให้กระบวนการคัดแยกแพ็กเก็ตกระทำพร้อมกันสองค่า โดยประมวลผลการค้นหาค่าพรีฟิกทั้งค่าพรีฟิกไอพีต้นทางและพรีฟิกไอพีปลายทางพร้อมกัน ยกตัวอย่างการคัดแยกแพ็กเก็ตตามโครงสร้างข้อมูลในภาพที่ 3 สมมติแพ็กเก็ต  $P$  {10101101, 11001010, 64, 1024, TCP} วิ่งเข้ามาที่อุปกรณ์จัดเส้นทาง  $R$  ซึ่งมีนโยบายของการคัดแยกแพ็กเก็ตโดยอิงตามค่าความสำคัญที่สูงที่สุด กระบวนการคัดแยกแพ็กเก็ตจะกระทำกับแพ็กเก็ต  $P$  แบบสองมิติโดยเริ่มจากการจับคู่แบบยาวที่สุด จับคู่กับค่าฟิลด์ {10101101, 11001010} ทั้งพรีฟิกต้นทางและพรีฟิกปลายทางซึ่งค่าพรีฟิกที่จับคู่กับ  $P$  ในแต่ละมิติจะได้เป็น 1) ค่าพรีฟิกไอพีต้นทางที่จับคู่จะได้ \* เพียงค่าเดียว 2) ค่าพรีฟิกไอพีปลายทาง ที่จับคู่ จะได้ \*, 1100\* และ 11001\* เมื่อ

นำค่าที่ จับคู่ แต่ละมิติมา Cross กัน ทำให้ได้ค่าจับคู่ทั้งสองมิติ คือ ค่าที่สอดคล้องกับช่องในตารางที่เท่ากับ 1 ในตาราง Cross-Product ซึ่งจากตัวอย่างนี้ ได้แก่ ค่าที่ พรีฟิกต้นทางเท่ากับ \* และพรีฟิกปลายทางเท่ากับ 1100\* จากนั้นจะใช้ค่า Code Word จาก Cross-Product นั่นคือค่า CW = 0 เพื่อใช้เป็นค่า Index ในแถวลำดับตัวกรองนั่นคือตัวกรองที่จับคู่กับค่า Cross-Product (\*, 1100\*) คือ  $F_8$  นอกจากนี้ ช่องในตาราง Cross-Product ที่เป็น 1 ที่สอดคล้องกับ  $P$  คือค่าพรีฟิกปลายทางเท่ากับ 11001\* อย่างไรก็ตามค่าช่องในตารางที่เป็น 1 นี้อยู่ในตำแหน่งที่ 2 ถัดจากตำแหน่งแรกของการจับคู่ ครั้งแรก เพราะฉะนั้นจะได้ค่า CW = 0 + 1 = 1 นั่นคือ Index ของแถวลำดับตัวกรองเป็น 1 และตัวกรองที่จับคู่นี้คือ  $F_4$  และสุดท้ายเมื่อพิจารณาที่ตัวกรองที่มีค่าความสำคัญสูงสุดจะได้ว่า  $F_4$  เป็นตัวกรองที่ดีที่สุดที่จับคู่กับ  $P$

		Group of destination prefixes						CW	Filter array (Highest priority)	
		*	1*	0111*	1100*	11001*	011110*			
	*	0	0	0	1	1	0	0	1	<b>F8</b>
	001*	0	0	1	0	0	0	2	3	F5
	0100*	0	1	0	0	0	0	3	4	F3
	01011*	0	0	0	0	0	1	4	5	F6
	10110*	0	0	0	1	0	0	5	6	F1
	001101*	1	0	0	0	0	0	6	7	F0

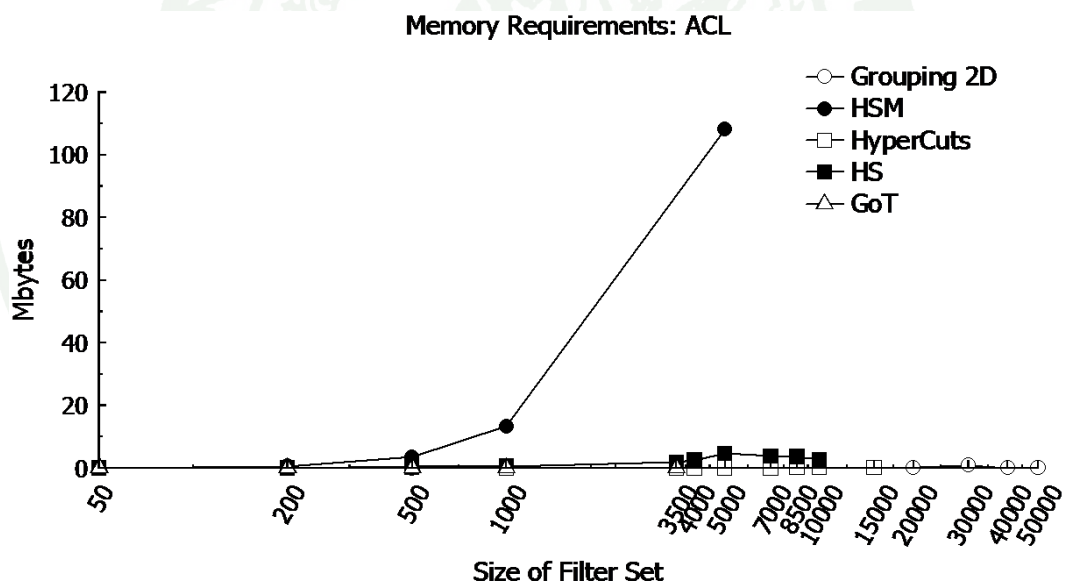
Matched filters!

ภาพที่ 4 ตัวอย่างการคัดแยกไอพีต่อโครงสร้างงานวิจัยที่น่าเสนอ

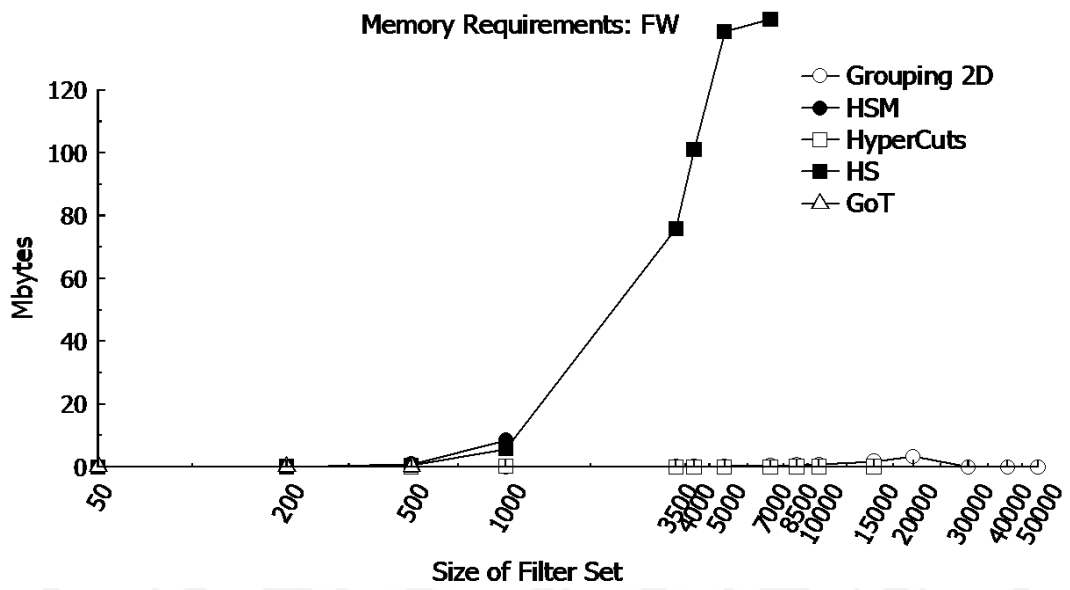
## ผลและวิจารณ์

จากที่กล่าวมาแล้วในเรื่องการวัดประสิทธิภาพขั้นตอนวิธี ผู้วิจัยจะใช้เกณฑ์การวัด 2 เกณฑ์ คือ การวัดขนาดหน่วยความจำที่ใช้ในโครงสร้างข้อมูลและจำนวนครั้งในการอ่านหน่วยความจำในกระบวนการค้นหาพรีฟิกโดยในการทดสอบนี้ ผู้วิจัยจะทดสอบผลโดยเปรียบเทียบกับงานวิจัย Grid of Trie (GoT) (Srinivasan *et al.*, 1998), Hypercuts (Singh *et al.*, 2003), HyperSplit (Qi *et al.*, 2009), HSM (Xu *et al.*, 2005) โดยใช้ชุดข้อมูลชุดตัวกรองที่ได้จาก ClassBench โดยชุดตัวกรองที่ใช้แบ่งออกเป็น 3 ประเภทคือ ACL, FW และ IPC โดยมีขนาดของชุดตัวกรองที่แตกต่างกันตั้งแต่ 50 ถึง 50,000 ตัว และรวมถึงผู้วิจัยได้ใช้ชุดข้อมูลแพ็กเก็ต (Trace) จาก ClassBench ซึ่งเป็น Generator จำลองแพ็กเก็ตเพื่อใช้ในการทดสอบกระบวนการคัดแยกพรีฟิก

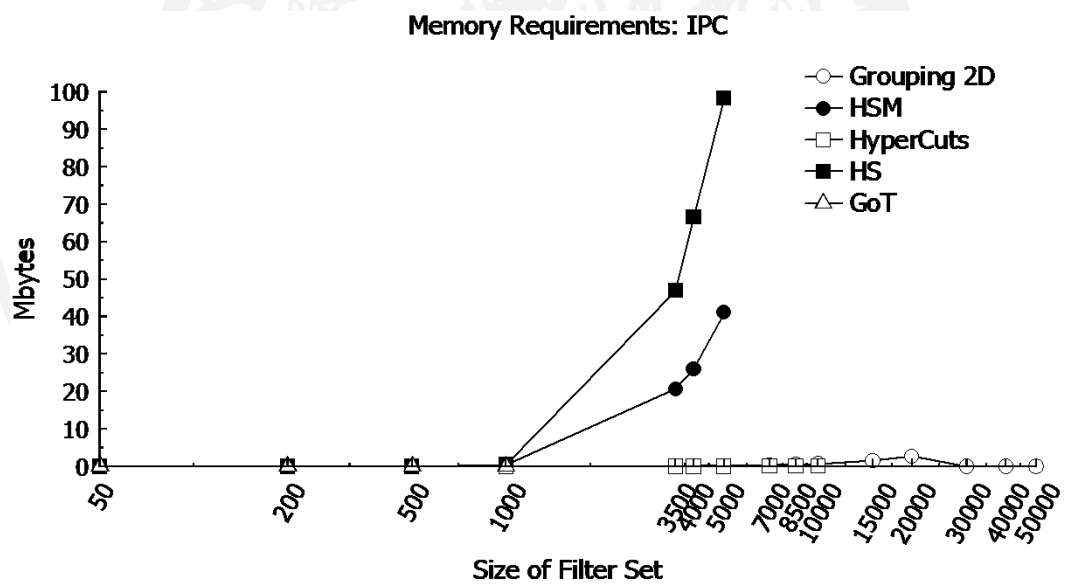
### ขนาดหน่วยความจำที่ใช้ในโครงสร้างข้อมูล



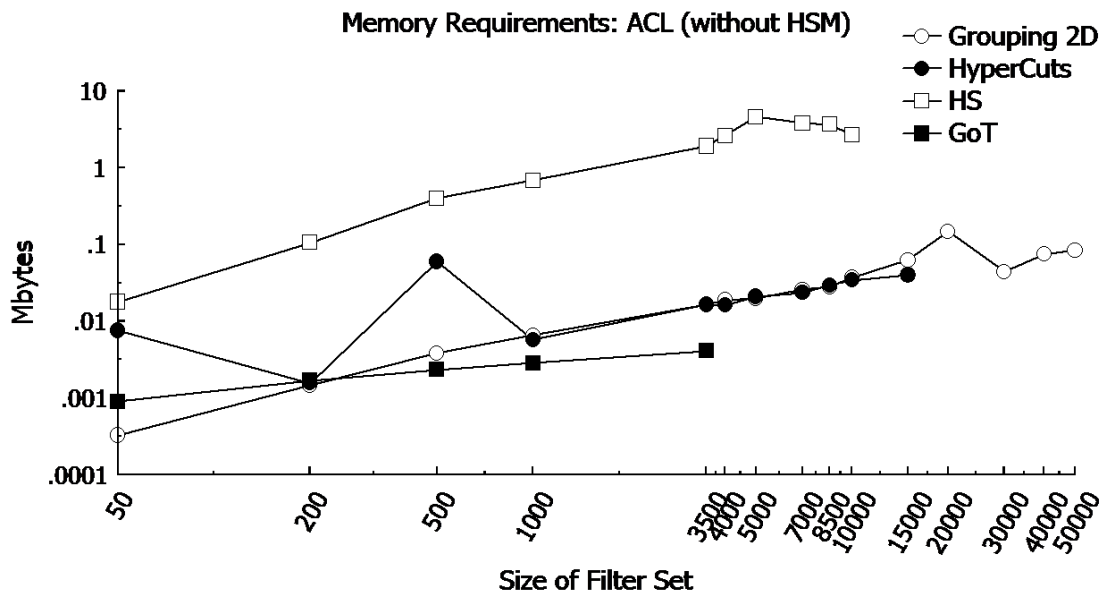
ภาพที่ 5 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL



ภาพที่ 6 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW



ภาพที่ 7 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด IPC



ภาพที่ 8 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองโดยไม่รวม HSM

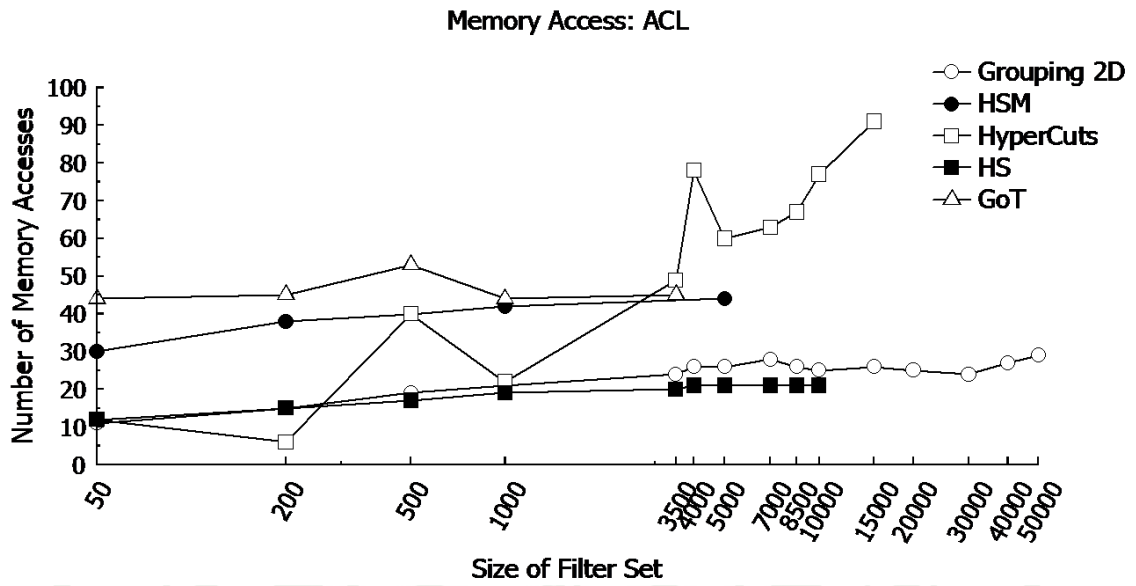
เนื่องจากหลักการสร้างโครงสร้างข้อมูลในงานวิจัยนี้มีการจัดตัวกรองออกเป็นกลุ่ม และมีการปรับโครงสร้างข้อมูล Cross-Product ให้เป็นแถวลำดับของบิต ดังนั้น ขนาดของหน่วยความจำที่ใช้สำหรับโครงสร้างข้อมูลแบบ Cross-Product นี้มีค่าน้อยมาก ทำให้ขนาดหน่วยความจำโดยรวมที่ใช้มีค่าน้อยมาก โดยเมื่อเทียบกับโครงสร้างข้อมูลอื่น ๆ ที่ใช้ชุดข้อมูลชุดตัวกรองเดียวกัน พบว่าโครงสร้างข้อมูลอื่น ๆ มีขนาดของหน่วยความจำมากกว่ามากตามแต่ละโครงสร้างนั้น ๆ จากภาพที่ 5-7 แสดงผลของขนาดหน่วยความจำที่ใช้ในโครงสร้างข้อมูล เมื่อพิจารณาตามขนาดตัวกรองพบว่าแต่ละโครงสร้างข้อมูลจะมีขนาดหน่วยความจำเพิ่มขึ้นอย่างมีนัยสำคัญ ได้แก่ งานวิจัย HSM และ HS ใช้หน่วยความจำเพิ่มขึ้นอย่างมากเมื่อขนาดตัวกรองมีค่าเพิ่มขึ้น โดยลักษณะกราฟจะเพิ่มขึ้นแบบ Exponential เช่น ชุดตัวกรองประเภท IPC ในภาพที่ 7 งานวิจัย HS ใช้หน่วยความจำสูงถึง 100 MBytes ส่วนงานวิจัย GoT และ HyperCuts แม้จะใช้หน่วยความจำไม่มากเมื่อเทียบกับ HSM และ HS แต่เห็นได้ว่า GoT และ HyperCuts จะไม่มีข้อมูลผลการทดสอบที่หน่วยความจำที่สูงขึ้น เช่น ภาพที่ 5 ตัวกรองประเภท ACL งานวิจัย GoT สามารถสร้างโครงสร้างข้อมูลได้แค่ชุดตัวกรองขนาด 1,000 ตัว และไม่สามารถสร้างได้มากกว่านี้ หรือภาพที่ 7 ตัวกรองประเภท IPC งานวิจัย HyperCuts ก็ไม่สามารถสร้างโครงสร้างข้อมูลจากชุดตัวกรองที่มีขนาดมากกว่า 8,500 ตัว ได้จากการทดสอบสามารถวิเคราะห์ได้ว่า GoT และ HyperCuts ไม่สามารถสร้างโครงสร้างข้อมูลด้วยชุดตัวกรองที่มีขนาดใหญ่ได้ ด้วยเหตุนี้ จึงไม่สามารถนำข้อมูลมาวิเคราะห์ได้ อย่งไรก็ดี เมื่อพิจารณาประเภทของตัวกรองพบว่าส่งผลต่อโครงสร้างข้อมูล เช่น ตัวกรองประเภท FW มีผลต่อ HS ที่ไม่สามารถสร้างโครงสร้างข้อมูลที่ชุดตัวกรองมีจำนวนตัวกรองมากกว่า 7,000 ตัว หรือตัวกรองประเภท IPC ที่ HSM ไม่สามารถสร้างโครงสร้างข้อมูลได้ตั้งแต่ ชุดตัวกรองขนาด 5,000 ตัว เป็น

ต้นไป ในส่วนนี้สามารถวิเคราะห์ได้ว่า ตัวกรองแต่ละประเภทจะมีรูปแบบค่าข้อมูลส่วนหัวแตกต่างกัน โดยเฉพาะค่าเลขที่อยู่ไอพีซึ่งมีผลเฉพาะต่อโครงสร้างข้อมูลประเภท Trie ที่เมื่อความยาวของพรีฟิกกระจายตัวมากเท่าใด ลักษณะโครงสร้างข้อมูลประเภท Trie จะซับซ้อนมากเท่านั้น (Taylor and Turner, 2005b) ทั้งนี้เมื่อพิจารณาการจัดกลุ่มสองมิติพบว่าทั้งจำนวนตัวกรองหรือประเภทตัวกรอง ไม่ส่งผลใด ๆ ต่อโครงสร้างข้อมูลเลย รวมทั้งขนาดโครงสร้างข้อมูลมีค่าน้อยมากซึ่งไม่เกิน 10 MBytes ซึ่งส่งผลให้การจัดกลุ่มสองมิติมีคุณสมบัติในเรื่องของ Scalability ต่อขนาดข้อมูลตัวกรองมากกว่างานวิจัยอื่น ๆ

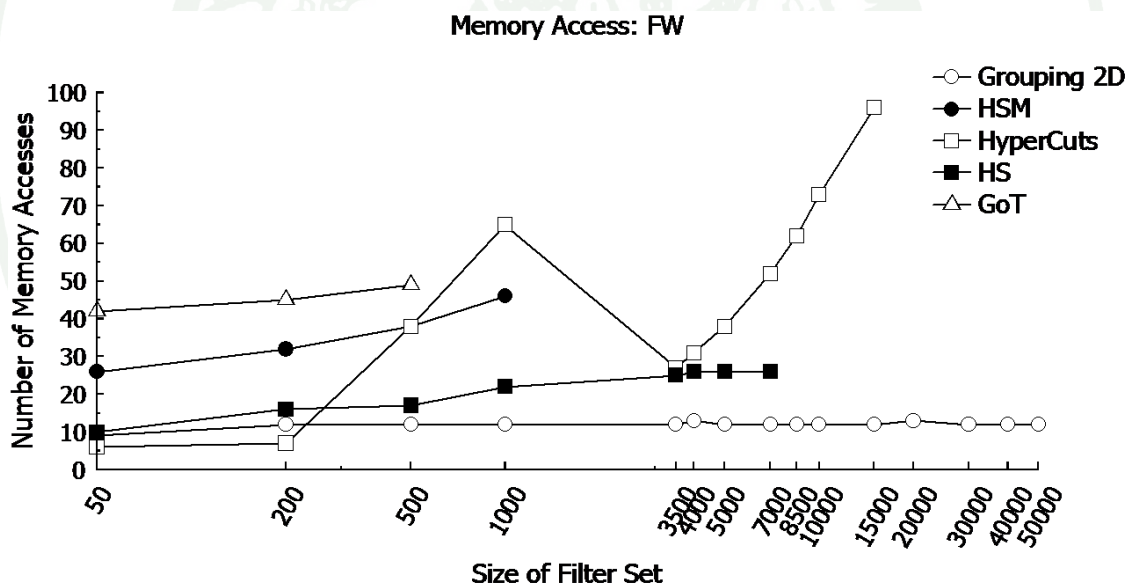
จากภาพที่ 5 ถ้าไม่พิจารณาขนาดหน่วยความจำที่ใช้ในโครงสร้างข้อมูลของ HSM จะได้กราฟดังภาพที่ 8 ซึ่งแสดงให้เห็นว่าทุกงานวิจัยมีอัตราการเพิ่มขึ้นของการใช้พื้นที่หน่วยความจำแบบเชิงเส้นตามขนาดตัวกรองที่เพิ่มขึ้น อย่างไรก็ตามการจัดกลุ่มสองมิติเป็นเพียงงานวิจัยเดียวที่สามารถสร้างโครงสร้างข้อมูลได้ถึง 50,000 ตัว ในขณะที่ใช้หน่วยความจำสูงสุดไม่ถึง 1 MBytes และการจัดกลุ่มสองมิติมีแนวโน้มที่สามารถรองรับการใช้งานได้เมื่อตัวกรองมีขนาดมากกว่า 50,000 ตัว

### จำนวนครั้งในการอ่านหน่วยความจำ

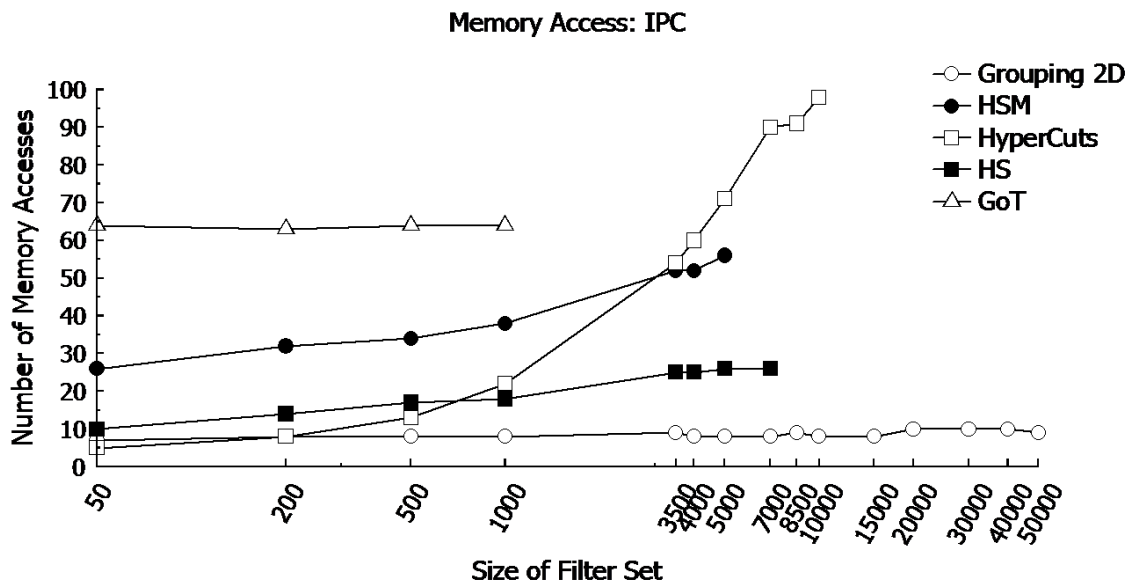
ในการนับจำนวนครั้งในการอ่านหน่วยความจำในกระบวนการค้นหาพรีฟิกจะนับทุก ๆ กระบวนการที่ CPU มีการอ่านค่าในหน่วยความจำ สำหรับงานวิจัยนี้จะมีการนับการเข้าถึงหน่วยความจำโดยแบ่งออกเป็น 3 ส่วน คือ เริ่มจากจับคู่แบบยาวที่สุดที่ใช้ขั้นตอนวิธี Helix เพื่อให้ได้ค่าตำแหน่งแวนอนในตาราง Cross-Product ซึ่งกระบวนการนี้ประมวลผลด้วยวิธี Parallel Computing ทั้งพรีฟิกต้นทางและปลายทาง จากงานวิจัย Helix จะได้ว่าในกระบวนการนี้จะนับค่าการเข้าถึงหน่วยความจำเป็น 1 หลังจากนั้นจะนับการเข้าถึงข้อมูลจากกระบวนการ Cross-Product โดยจะนับตามจำนวนแถวแวนอนที่ได้จากงานวิจัย Helix ซึ่งการเข้าถึงแถวลำดับของบิต ในตาราง Cross-Product สามารถแทนด้วยการนับค่าจำนวน Word ได้ (1 Word คือ 16 บิต) และการนับสุดท้าย คือ การนับจำนวนครั้งในการอ่านข้อมูลในแถวลำดับตัวกรองโดยกำหนดให้แต่ละช่องในแถวลำดับตัวกรองเก็บเลขที่ตัวกรอง (Filter Number) ที่มีความสำคัญสูงที่สุดแค่ 1 ค่าเท่านั้น เพราะฉะนั้นจำนวนครั้งในการอ่านหน่วยความจำจะนับเป็น 1 เสมอ



ภาพที่ 9 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด ACL



ภาพที่ 10 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด FW



ภาพที่ 11 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด IPC

จากการทดสอบทำให้ทราบว่า ในการคัดแยกไอพีแบบการจัดกลุ่มสองมิติ จะได้จำนวนครั้งที่อ่านข้อมูลจากหน่วยความจำโดยรวมมีค่าน้อยมากเมื่อเทียบกับขั้นตอนวิธีอื่น ๆ ดังภาพที่ 9-11 เมื่อพิจารณาผลของแต่ละงานวิจัย พบว่าจำนวนตัวกรองส่งผลต่อประสิทธิภาพการคัดแยกไอพีต่องานวิจัย HyperCuts มากที่สุด โดยเมื่อจำนวนตัวกรองมีขนาดใหญ่ขึ้น จำนวนครั้งที่อ่านข้อมูลจากหน่วยความจำต่อการคัดแยกไอพี 1 ครั้งมีค่าเพิ่มขึ้น สำหรับงานวิจัย HSM และ HS จำนวนครั้งในการเข้าถึงข้อมูลจะค่อย ๆ เพิ่มขึ้นสอดคล้องกับจำนวนตัวกรองที่เพิ่มขึ้นแต่ไม่มากเท่า HyperCuts โดยลักษณะการเพิ่มขึ้นจะอยู่ในรูปเชิงเส้น ส่วนงานวิจัย GoT จะให้ผลจำนวนครั้งที่อ่านข้อมูลจากหน่วยความจำไม่เกิน 64 ครั้งสอดคล้องกับลักษณะโครงสร้างข้อมูลแบบ Trie สองมิติที่มีค่าความสูงของ Trie เป็นค่า 64 อย่างไม่รู้ก็ตาม งานวิจัยที่นำเสนอมีอัตราการเพิ่มขึ้นช้าที่สุดเมื่อเทียบกับงานวิจัยอื่น ๆ โดยลักษณะการเพิ่มขึ้นแทบจะคงที่โดยเฉพาะเมื่อพิจารณาที่ตัวกรองประเภท FW และ IPC ซึ่งสามารถวิเคราะห์ได้ว่าโครงสร้างข้อมูลแบบการจัดกลุ่มสองมิติ เป็นแถวลำดับของบิตที่ได้จาก Cross-product ระหว่างพรีฟิกไอพีต้นทางและปลายทาง ซึ่งเป็นโครงสร้างข้อมูลที่มีขนาดเล็กที่สุดเมื่อเทียบกับงานวิจัยอื่น ส่งผลให้การคัดแยกไอพีมีความรวดเร็วตามลักษณะโครงสร้างข้อมูลด้วย นอกจากนี้สามารถวิเคราะห์ได้ว่าขนาดตัวกรองไม่มีนัยสำคัญต่อจำนวนครั้งในการเข้าถึงหน่วยความจำต่อการคัดแยกไอพีดังภาพที่ 9-11 จำนวนครั้งในการเข้าถึงหน่วยความจำของการจัดกลุ่มสองมิตินี้มีค่าคงที่ในช่วง 10 ถึง 20 ครั้งในตัวกรองขนาด 50 ถึง 50,000 ตัว

จากผลการทดสอบการเข้าถึงหน่วยความจำ สามารถคิดเป็นอัตราการคัดแยกแพ็กเก็ตได้  
ดังนี้ ถ้าหน่วยความจำ SRAM ซึ่ง CPU ใช้เวลาในการเข้าถึง 1 Word เป็น 10 Nanoseconds (Intel)  
จากผลการทดสอบของงานวิจัยนี้ในภาพที่ 9 พบว่า การคัดแยกแพ็กเก็ตเกิดของชุดตัวกรองแบบ ACL  
ขนาด 50,000 ตัว ใช้จำนวนครั้งการเข้าถึงหน่วยความจำ 20 ครั้งต่อการคัดแยก 1 แพ็กเก็ต ดังนั้น  
เวลาที่ใช้การคัดแยก 1 แพ็กเก็ตคิดเป็น 200 Nanoseconds หรือ 0.0002 Milliseconds หรือกล่าวได้  
ว่ามีอัตราการคัดแยก 5 ล้านไอพีแพ็กเก็ตต่อ 1 วินาที ซึ่งถือว่าเร็วมาก



## สรุปและข้อเสนอแนะ

การพัฒนาขั้นตอนวิธีคัดแยกแพ็กเก็ตแบบสองมิติที่ใช้หน่วยความจำ SRAM ยังคงมีความสำคัญมากกว่าเมื่อเทียบกับการพัฒนา TCAMs อันเนื่องมาจากข้อจำกัดเกี่ยวกับขั้นตอนวิธีคัดแยกแพ็กเก็ตและข้อจำกัดทางด้านพลังงานและค่าใช้จ่ายในการใช้งาน TCAMs และการคัดแยกแพ็กเก็ตแบบสองมิติก็ยังคงมีความสำคัญต่องานในระบบเครือข่าย ซึ่งการพัฒนาการคัดแยกแพ็กเก็ตนี้ยังได้รับความสำคัญและมีการพัฒนาเรื่อยมา สำหรับขั้นตอนวิธีการคัดแยกแพ็กเก็ตแบบสองมิติ งานวิจัยนี้ได้แสดงให้เห็นว่า มีประสิทธิภาพมากทั้งในแง่ของความเร็วในการคัดแยกไอพี อันเนื่องมาจากแนวคิดในการสร้างโครงสร้างข้อมูลโดยการจัดกลุ่มฟิลด์พรีฟิกไอพีต้นทางและปลายทาง ซึ่งจากการทดสอบประสิทธิภาพขั้นตอนวิธี (Algorithm) ของงานวิจัยนี้มีความเร็วมากที่สุดเมื่อเทียบกับงานวิจัยอื่น ๆ โดยเฉพาะเมื่อคิดเป็นอัตราการจัดแยกเลขที่อยู่ไอพี มีความเร็วมากกว่าถึง 6 เท่า นอกจากนี้การใช้พื้นที่หน่วยความจำของโครงสร้างข้อมูลเมื่อเปรียบเทียบกับงานวิจัยอื่น ๆ แล้วก็มีค่าน้อยกว่าถึง 10 เท่า รวมทั้งโครงสร้างข้อมูลนี้ยังมีความสามารถในการรองรับชุดตัวกรองที่มีขนาดใหญ่กว่านี้มาก ๆ ได้ (Scalability) เพราะ การจัดกลุ่มข้อมูลส่วนหัวก่อนที่จะนำข้อมูลมาสร้าง โครงสร้างข้อมูล (Pre-Computed Data) ซึ่งสมเหตุสมผลที่ว่าถึงแม้ชุดตัวกรองที่มีขนาดใหญ่ เมื่อจัดกลุ่มค่าความเป็นไปได้โดยตรงแล้วก็จะได้จำนวนตัวกรองที่แท้จริงน้อยมาก (แต่ค่าเลขตัวกรองยังคงเท่าเดิม) จากการทดสอบข้างต้น ทำให้งานวิจัยนี้มีประสิทธิภาพที่เหมาะสมที่สุดที่จะนำมาใช้งานจริง เพราะด้วย โครงสร้างข้อมูลที่ไม่ซับซ้อน และสามารถตอบสนองต่อการคัดแยกแพ็กเก็ตได้อย่างรวดเร็ว ซึ่งจะช่วยลดปัญหาคอขวดที่เกิดกับอุปกรณ์จัดเส้นทางและสร้างความพึงพอใจแก่ผู้ใช้บริการอินเทอร์เน็ตได้เป็นอย่างดี

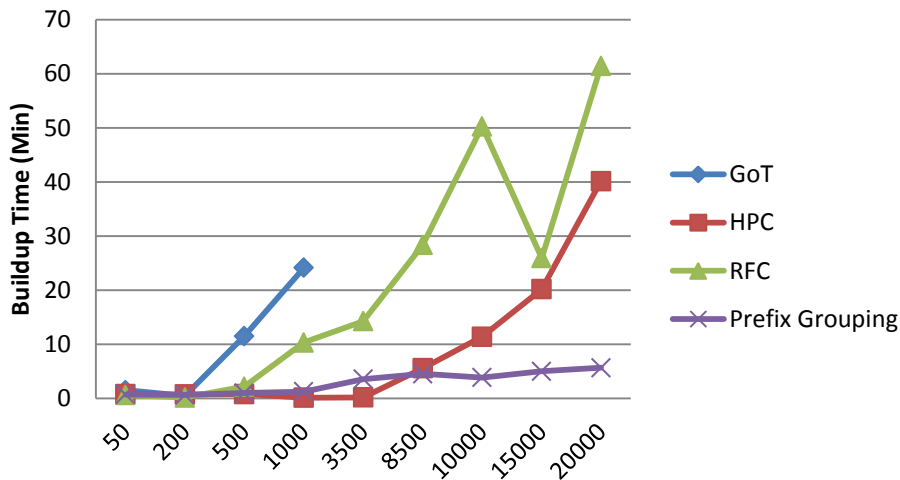
## งานวิจัยที่จะทำในอนาคต

ปัญหาการคัดแยกแพ็กเก็ตแบบสองฟิลด์ (พิจารณาที่ไอพีต้นทางและไอพีปลายทาง) นับเป็นส่วนหนึ่งของปัญหาการคัดแยกแพ็กเก็ตแบบห้าฟิลด์ (ฟิลด์อื่นๆ ได้แก่ หมายเลขพอร์ตต้นทาง, หมายเลขพอร์ตปลายทาง และ โพรโตคอล) ซึ่งงานวิจัย การจัดกลุ่มพรีฟิก นี้สามารถพัฒนาเพื่อนำไปประยุกต์ใช้กับการคัดแยกแพ็กเก็ตแบบห้าฟิลด์ได้ โดยพิจารณาดังนี้

1. โครงสร้างข้อมูลตาราง Cross-Product เกิดจากการนำค่ากลุ่มไอพีต้นทางและไอพีปลายทางมาสร้างเป็นโครงสร้างข้อมูล ซึ่งพอร์ตต้นทางและพอร์ตปลายทางสามารถนำมาประยุกต์ให้เข้ากับโครงสร้างตาราง Cross-Product ได้

2. เนื่องจากงานวิจัยนี้ใช้เวลาในการสร้างโครงสร้างข้อมูลไม่นานนักเพื่อเปรียบเทียบกับงานวิจัยอื่นๆ ดังภาพผนวกที่ 26 เพราะฉะนั้นเราสามารถพัฒนางานวิจัยให้ประยุกต์กับการคัดแยกแพ็กเก็ตแบบห้าฟิลด์ได้โดยง่าย

### Buildup Time



ภาพที่ 12 เวลาในการสร้างโครงสร้างข้อมูลในแต่ละชุดตัวกรอง (แสดงเป็นหน่วย นาที)

3. ลักษณะโครงสร้างข้อมูลไม่มีความซับซ้อนเพราะประกอบด้วยแถวลำดับแบบสองมิติสำหรับตาราง Cross-Product และแถวลำดับหนึ่ง มิติสำหรับฟิลเตอร์แถวลำดับ เพราะฉะนั้นเราสามารถนำโครงสร้างข้อมูลนี้มาติดตั้งบนฮาร์ดแวร์และใช้งานได้จริง

## เอกสารและสิ่งอ้างอิง

- Apipattanamongre, P. and T. Kijkanjanarat. January – April 2004. Fast Filter Matching in Two Dimensions by Using Trie-Based Structures. **Fast Filter Matching in Two Dimensions by Using Trie-Based Structures** 12 12-22.
- Baboescu, F. and G. Varghese. 2005. Scalable Packet Classification. **Scalable Packet Classification** 13 (1): 2-14.
- Baboescu, F., S. Sumeet and G. Varghese. 2003. **Packet Classification for Core Routers: Is There an Alternative to Cams?**, 53-63 vol.51. *In.*
- Dixit, M., B.V. Barbadekar and A.B. Barbadekar. 2009. Packet Classification Algorithms, น. 1407-1412. ใน **Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on.**
- Gupta, P. and N. McKeown. 1999a. Packet Classification on Multiple Fields. **Packet Classification on Multiple Fields** 29 (4): 147-160.
- Gupta, P. and N. McKeown. 1999b. **Packet Classification Using Hierarchical Intelligent Cuttings**, 34-41. *In.*
- Gupta, P. and N. McKeown. 2001. Algorithms for Packet Classification. **Algorithms for Packet Classification** 15 (2): 24-32.
- Intel. Available Source: <http://software.intel.com/en-us/articles/recap-virtual-memory-and-cache>
- Kempke, M. November, 1998. Ternary Cam Memory Architecture and Methodology. **Ternary Cam Memory Architecture and Methodology**

Montoye. June 1994. Don't Care" in a Content Addressable Memory Cell. **Don't Care" in a Content Addressable Memory Cell**

Qi, Y., L. Xu, B. Yang, Y. Xue and J. Li. 2009. **Packet Classification Algorithms: From Theory to Practice**, 648-656. *In.* IEEE,

Rojas-Cessa, R., T. Kijkanjanarat, N. Thirapittayatakul and P. Apipattanamontre. 2010. Practical and Scalable Method to Perform Ip Lookup in One Memory Access Time. **Practical and Scalable Method to Perform Ip Lookup in One Memory Access Time** Technical Report (THMNJ.TR201006), Dept. of Elect. and Comp. Eng., New Jersey Institute of Technology, Newark, NJ

Singh, S., F. Baboescu, G. Varghese and J. Wang. 2003. Packet Classification Using Multidimensional Cutting, u. 213-224. ใน **Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications**. ACM, Karlsruhe, Germany.

Srinivasan, V., G. Varghese, S. Suri and M. Waldvogel. 1998. Fast and Scalable Layer Four Switching. **Fast and Scalable Layer Four Switching** 28 (4): 191-202.

Taylor, D.E. 2005. Survey and Taxonomy of Packet Classification Techniques. **Survey and Taxonomy of Packet Classification Techniques** 37 (3): 238-275.

Taylor, D.E. and J.S. Turner. 2005a. **Scalable Packet Classification Using Distributed Crossproducing of Field Labels**, 269-280 vol. 261. *In.*

Taylor, D.E. and J.S. Turner. 2005b. **ClassBench: A Packet Classification Benchmark**, 2068-2079 vol. 2063. *In.*

Wencheng, L. and S. Sahni. 2006. **Packet Classification Using Pipelined Two-Dimensional Multibit Tries**, 808-813. *In.*

Woo, T.Y.C. 2000. A Modular Approach to Packet Classification: Algorithms and Results. **A Modular Approach to Packet Classification: Algorithms and Results**

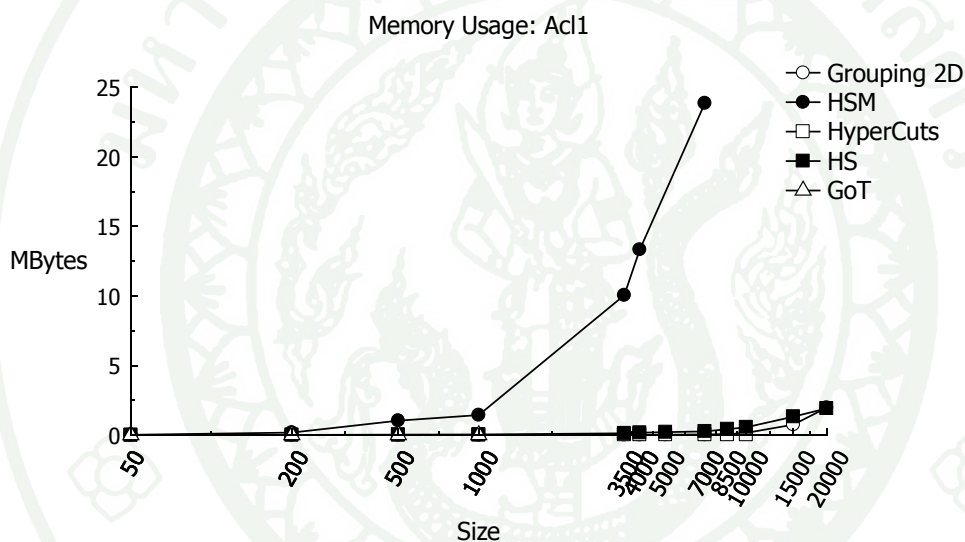
Xu, B., D. Jiang and J. Li. 2005. **Hsm: A Fast Packet Classification Algorithm**, 987-992. *In*. IEEE,



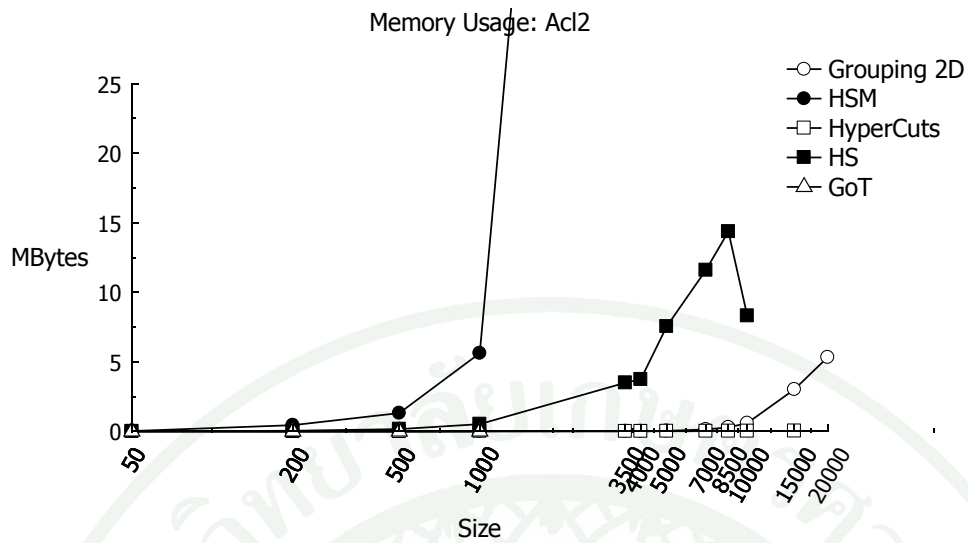




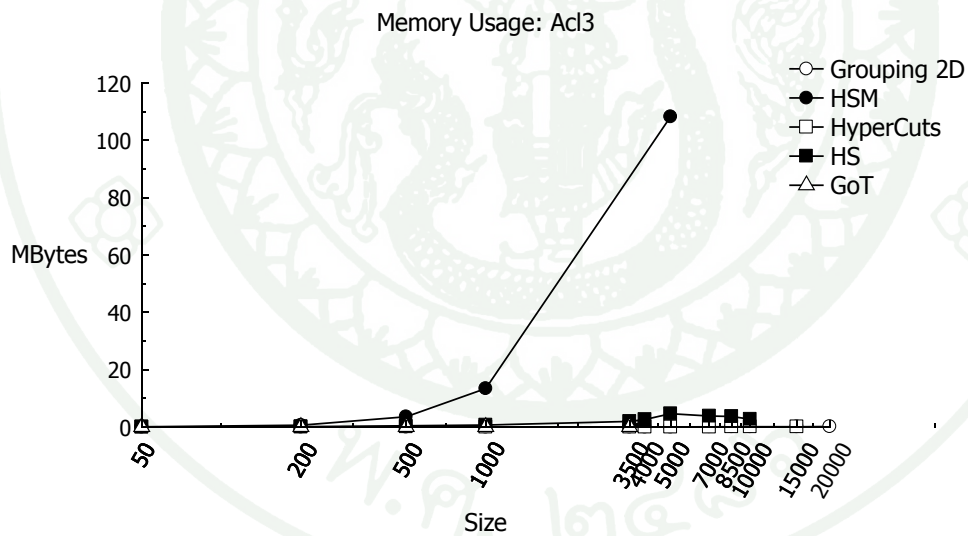
ในการวัดขนาดหน่วยความจำที่ใช้ในโครงสร้างข้อมูล ผู้วิจัยทดสอบผลโดยเปรียบเทียบกับงานวิจัย Grid of Trie, Hypercuts, HyperSplit และ HSM โดยใช้ชุดข้อมูลชุดตัวกรองที่ได้จาก ClassBench ซึ่งชุดตัวกรองที่ใช้แบ่งออกเป็นสามประเภทคือ ACL, FW และ IPC โดยชุดตัวกรองแต่ละประเภทแบ่งออกเป็นชุดตัวกรองย่อยคือ ACL1, ACL2, ACL3, ACL4, ACL5, FW1, FW2, FW3, FW4, FW5, IPC1 และ IPC2 และมีการแบ่งขนาดของชุดตัวกรองย่อยออกเป็นหลาย ๆ ขนาดคือ 50, 200, 500, 1,000, 3,500, 4,000, 5,000, 7,000, 8,500, 10,000, 15,000 และ 20,000 ตัว สำหรับผลการทดสอบหน่วยความจำที่ใช้ในโครงสร้างข้อมูลของตัวกรองแต่ละชนิด แสดงดังภาพผนวกที่ ก1 ถึง ก12



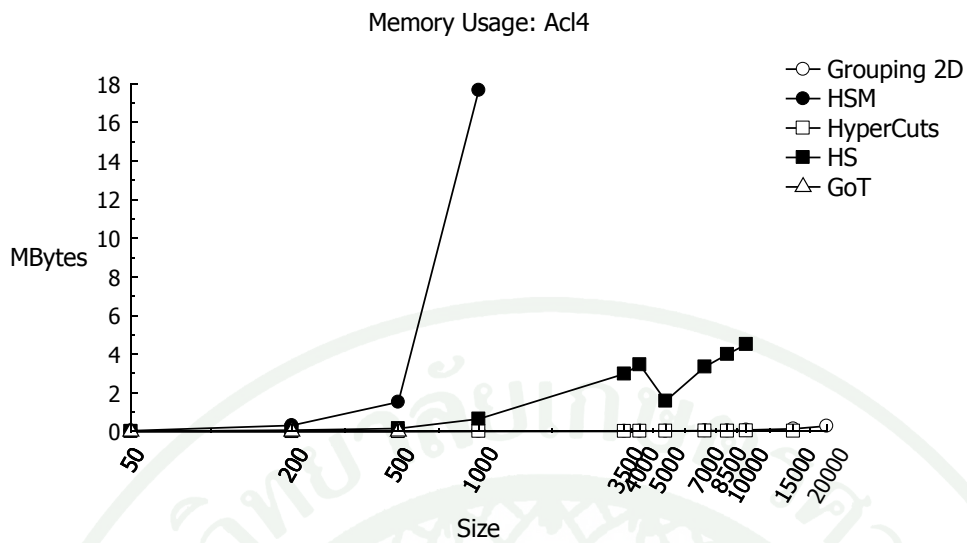
ภาพผนวกที่ ก1 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL1 จากภาพ HSM จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติใช้หน่วยความจำแค่ 2.5 MB สำหรับตัวกรอง 20,000 ตัว



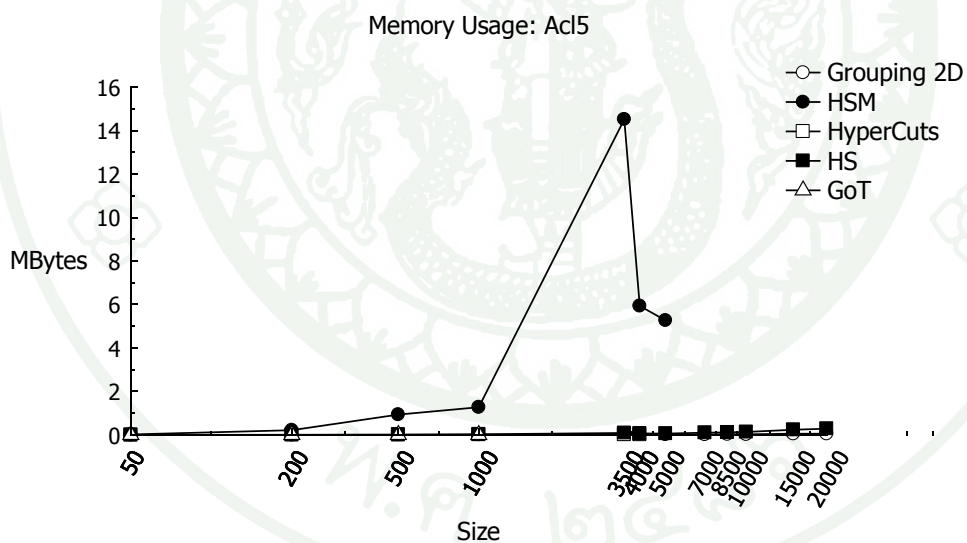
ภาพผนวกที่ ก2 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL2 จากภาพ HSM และ HS จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติใช้หน่วยความจำสูงสุด 5 MB



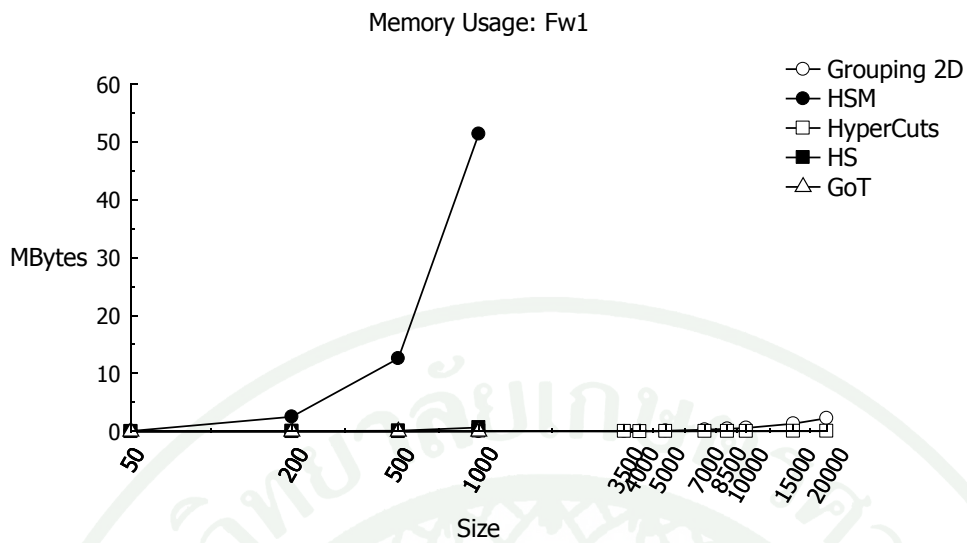
ภาพผนวกที่ ก3 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL3 จากภาพ HSM จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติ ใช้หน่วยความจำสูงสุดไม่ถึง 2 MB



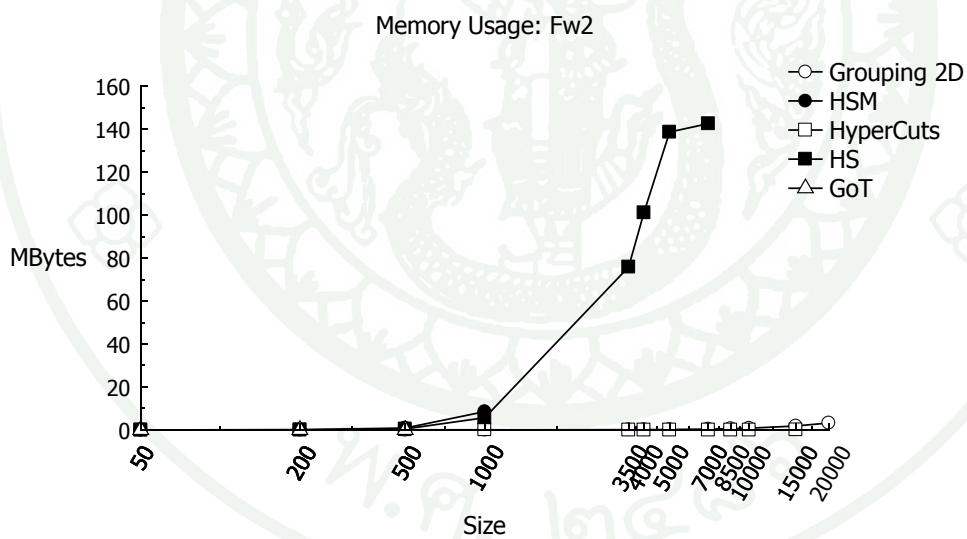
ภาพผนวกที่ ก4 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL4 จากภาพ HSM และ HS จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติ ใช้หน่วยความจำสูงสุดไม่ถึง 1 MB



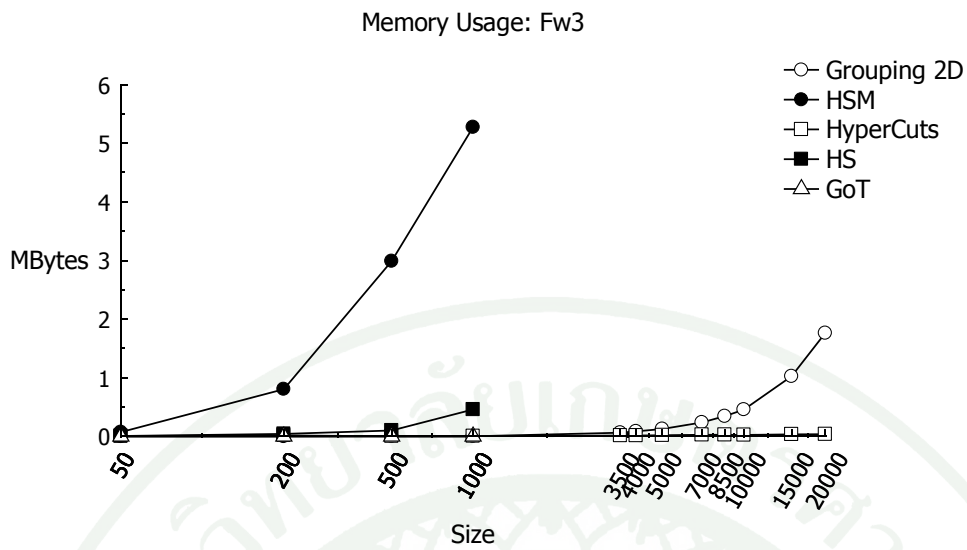
ภาพผนวกที่ ก5 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด ACL5 จากภาพ HSM จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุดเมื่อตัวกรองขนาด 3,500 ตัว และไม่สามารถสร้างโครงสร้างข้อมูลได้อีก



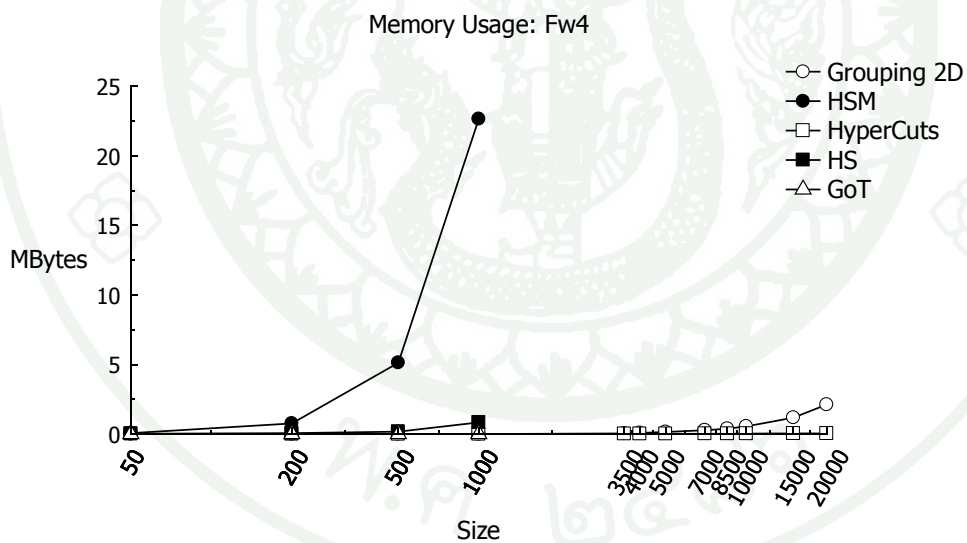
ภาพผนวกที่ 6 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW1 จากภาพ HSM จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติ ใช้หน่วยความจำแค่ 5 MB สำหรับตัวกรอง 20,000 ตัว



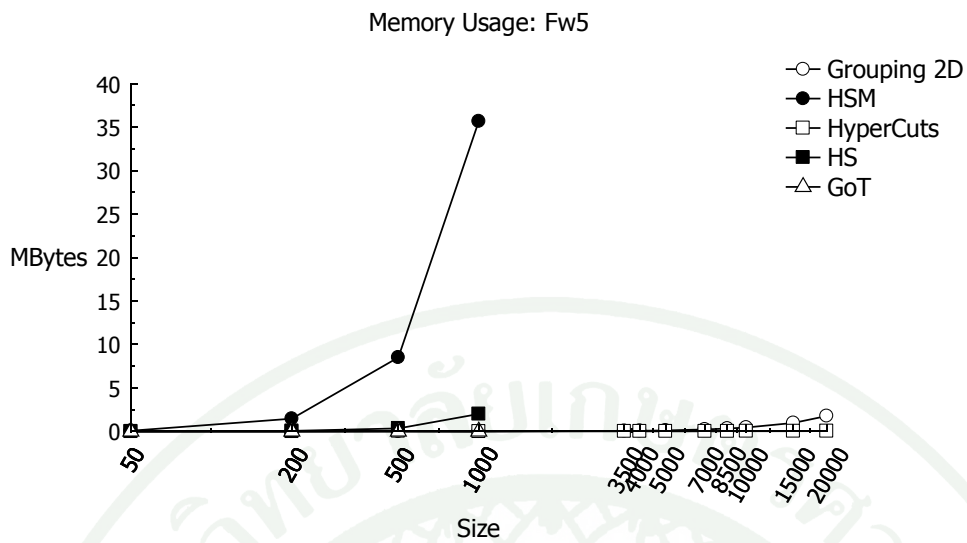
ภาพผนวกที่ 7 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW2 จากภาพ HS จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติ ใช้หน่วยความจำไม่ถึง 10 MB สำหรับตัวกรอง 20,000 ตัว



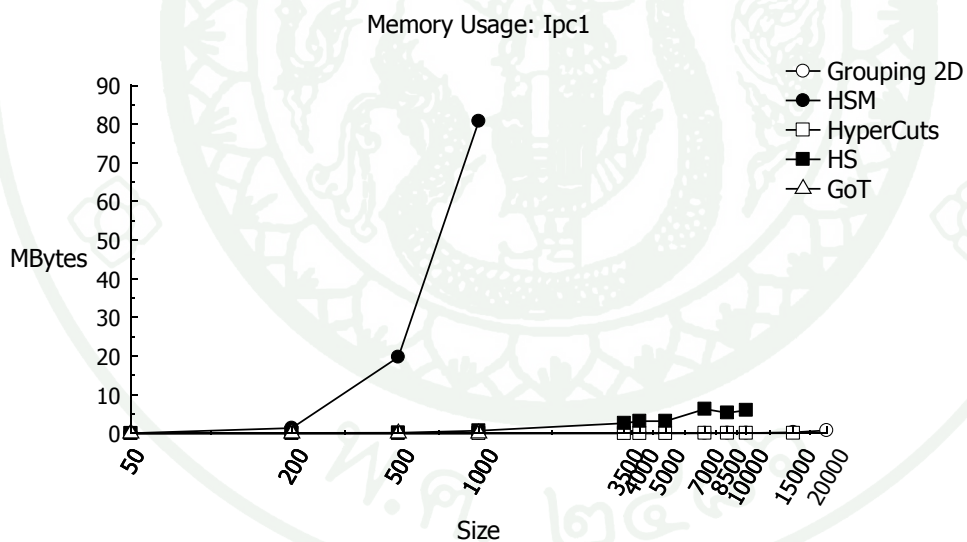
ภาพผนวกที่ ก8 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW3 จากภาพ HSM จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติใช้หน่วยความจำ 2 MB สำหรับตัวกรอง 20,000 ตัว



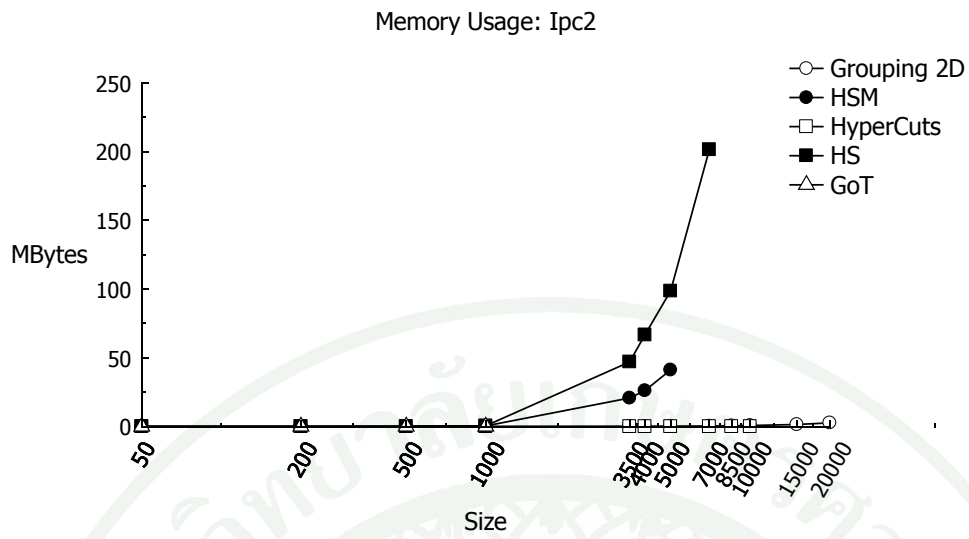
ภาพผนวกที่ ก9 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW4 จากภาพ HSM จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติใช้หน่วยความจำแค่ 2.5 MB สำหรับตัวกรอง 20,000 ตัว



ภาพผนวกที่ ก10 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด FW5 จากภาพ HSM จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติใช้หน่วยความจำสูงสุดไม่ถึง 2 MB



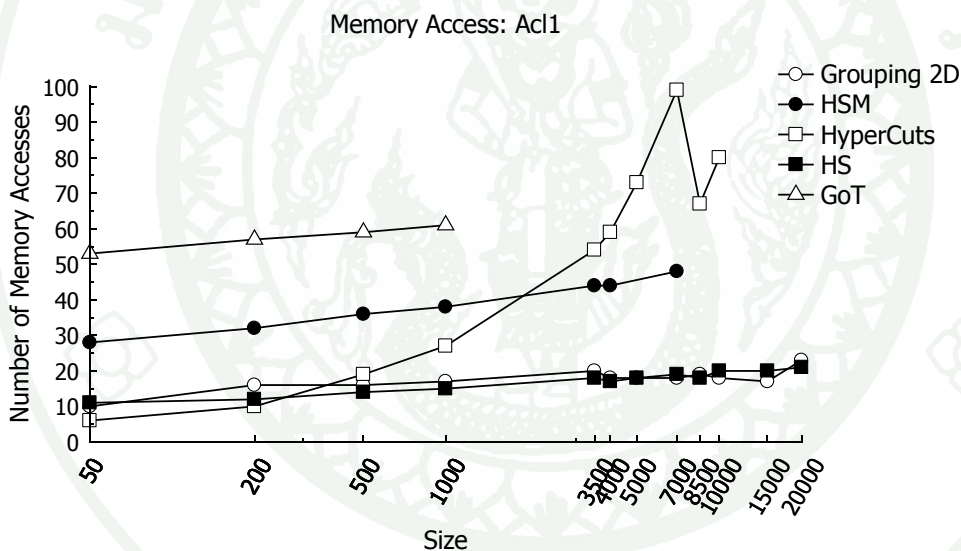
ภาพผนวกที่ ก11 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด IPC1 จากภาพ HSM จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติใช้หน่วยความจำน้อยมากสำหรับตัวกรอง 20,000 ตัว



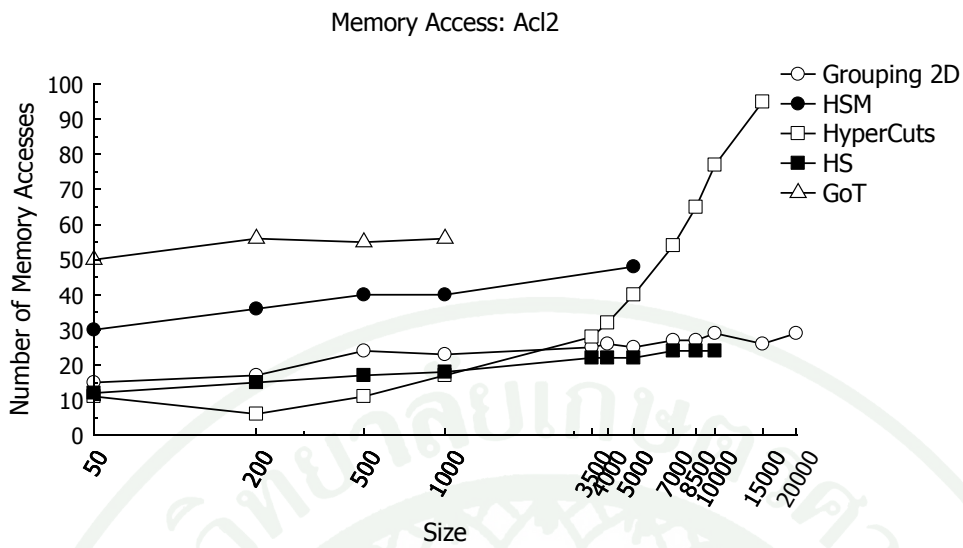
ภาพผนวกที่ ก12 ความต้องการพื้นที่หน่วยความจำสำหรับตัวกรองชนิด IPC2 จากภาพ HS และ HSM จะมีจำนวนการใช้พื้นที่หน่วยความจำสูงสุด



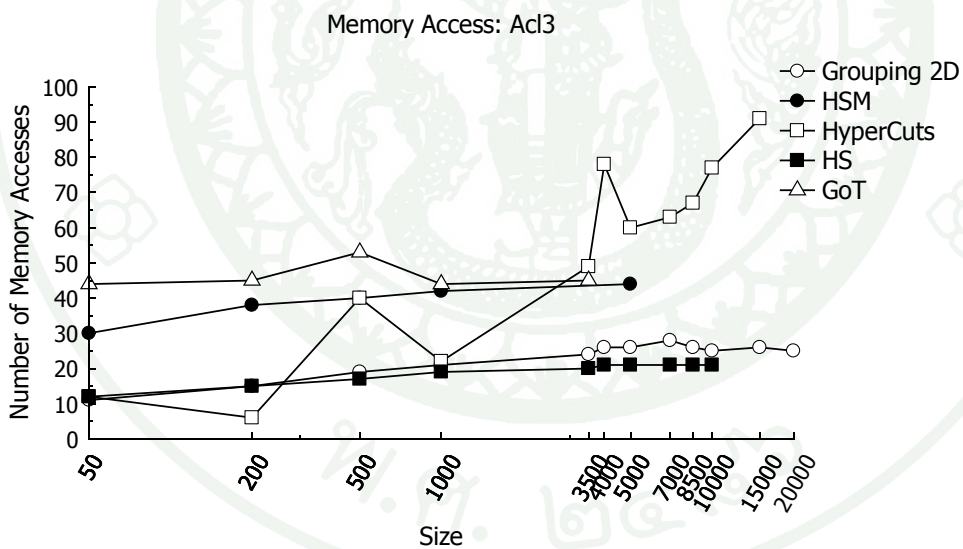
ในการวัดจำนวนครั้งในการอ่านหน่วยความจำของการคัดแยกแพ็กเก็ต ผู้วิจัยทดสอบผล โดยเปรียบเทียบกับงานวิจัย Grid of Trie, Hypercuts, HyperSplit และ HSM โดยใช้ข้อมูลชุดตัวกรองที่ได้จาก ClassBench ซึ่งชุดตัวกรองที่ใช้แบ่งออกเป็นสามประเภทคือ ACL, FW และ IPC โดยชุดตัวกรองแต่ละประเภทแบ่งออกเป็นชุดตัวกรองย่อยคือ ACL1, ACL2, ACL3, ACL4, ACL5, FW1, FW2, FW3, FW4, FW5, IPC1 และ IPC2 และมีการแบ่งขนาดของชุดตัวกรองย่อยออกเป็นหลาย ๆ ขนาดคือ 50, 200, 500, 1,000, 3,500, 4,000, 5,000, 7,000, 8,500, 10,000, 15,000 และ 20,000 ตัว และรวมถึงผู้วิจัยได้ใช้ชุดข้อมูลแพ็กเก็ต (Trace) จาก ClassBench ซึ่งให้ค่าแพ็กเก็ตจำลองเพื่อใช้ในการทดสอบกระบวนการคัดแยกฟรีฟิก จากการทดสอบทำให้ทราบว่า ในการคัดแยกไอพีแบบการจัดกลุ่มสองมิติ จะได้จำนวนครั้งที่ย่านข้อมูลจากหน่วยความจำโดยรวมมีค่าน้อยมากเมื่อเทียบกับขั้นตอนวิธีอื่น ๆ สำหรับผลการทดสอบจำนวนครั้งในการอ่านหน่วยความจำในการคัดแยกแพ็กเก็ตต่อตัวกรองแต่ละชนิดแสดงดังภาพผนวกที่ ข1 ถึง ข12



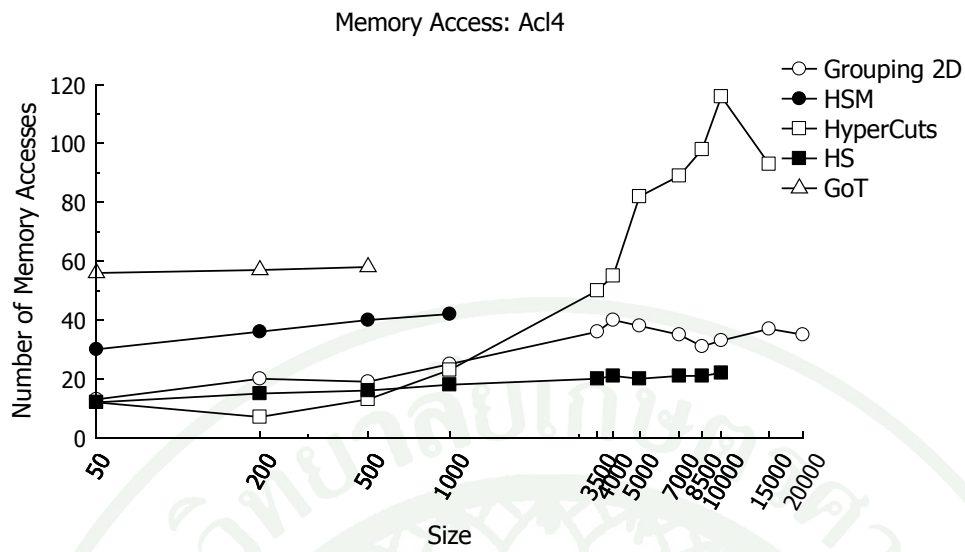
ภาพผนวกที่ ข1 การอ้างถึงหน่วยความจำสำหรับตัวกรองชนิด ACL1 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุด 20 ครั้ง



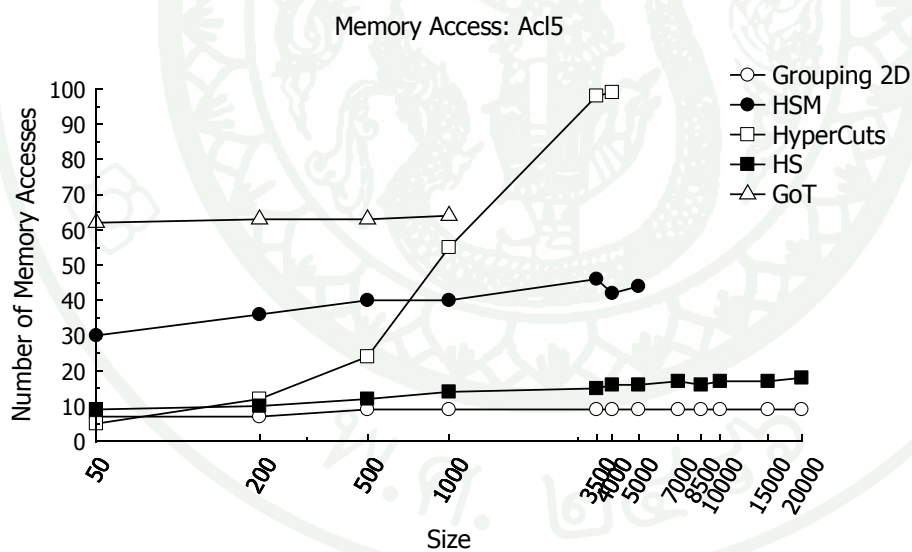
ภาพผนวกที่ ข2 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด ACL2 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุดไม่ถึง 30 ครั้ง



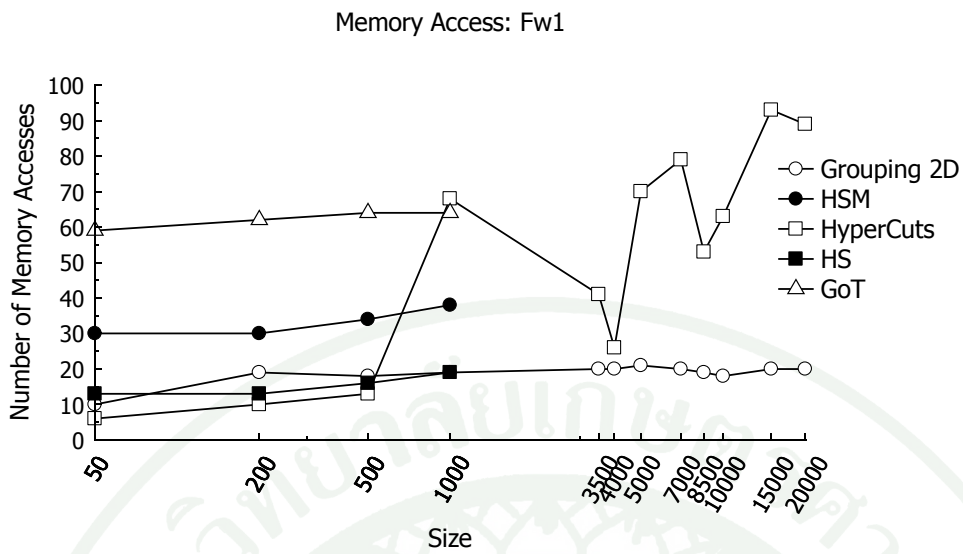
ภาพผนวกที่ ข3 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด ACL3 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุดไม่ถึง 30 ครั้ง



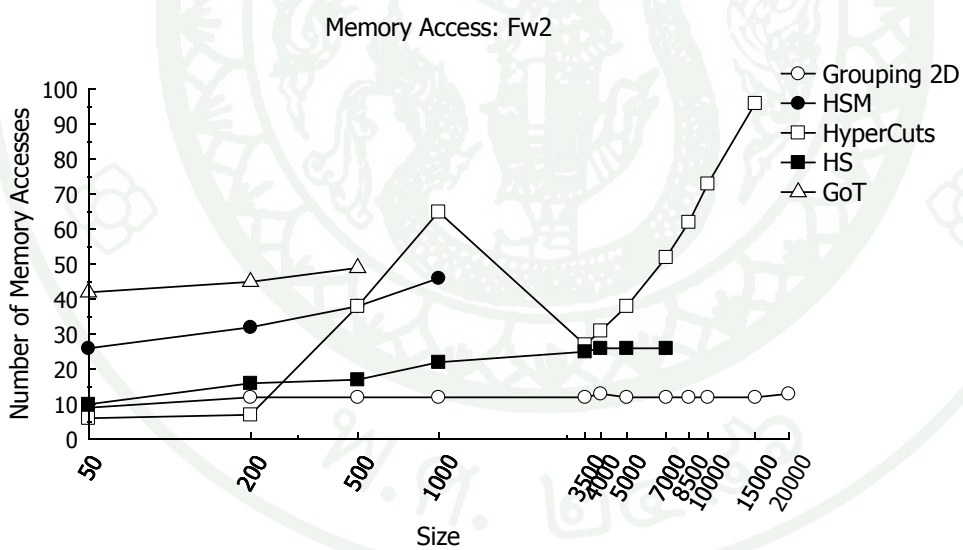
ภาพผนวกที่ ข4 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด ACL4 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุด 26 ครั้ง



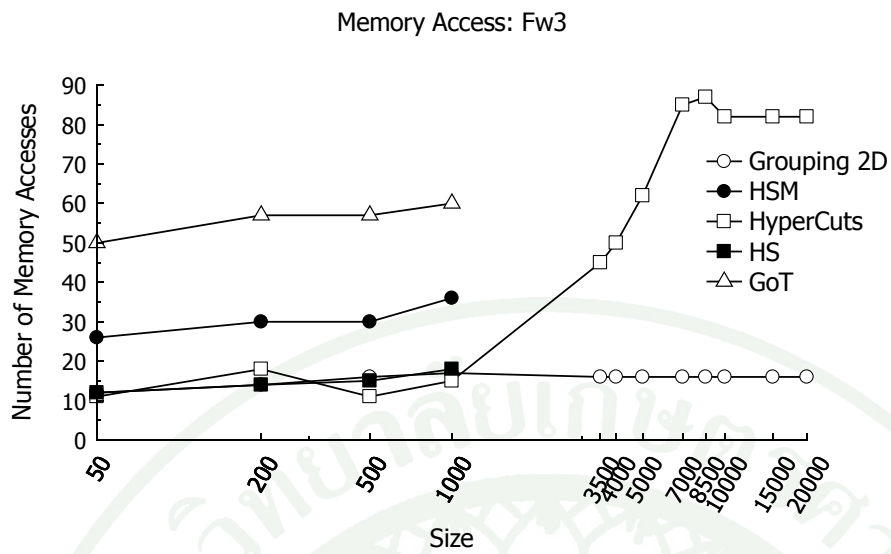
ภาพผนวกที่ ข5 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด ACL5 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุด 10 ครั้ง



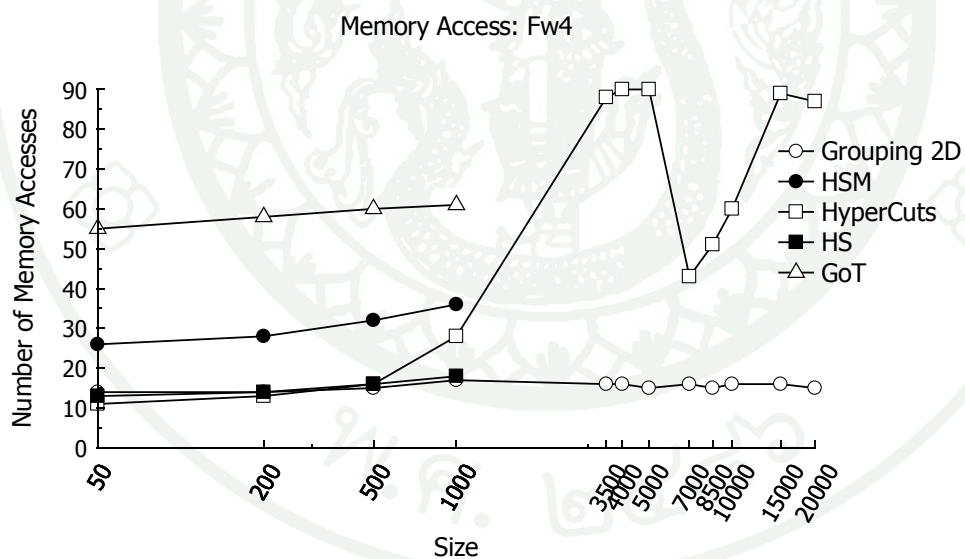
ภาพผนวกที่ ข6 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด FW1 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุด 20 ครั้ง



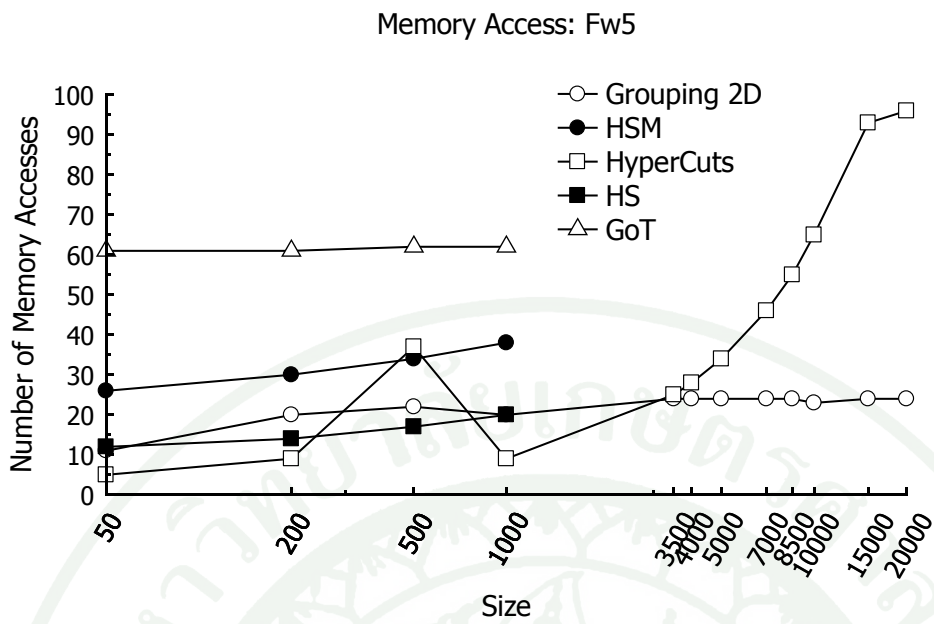
ภาพผนวกที่ ข7 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด FW2 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุด 10 ครั้ง



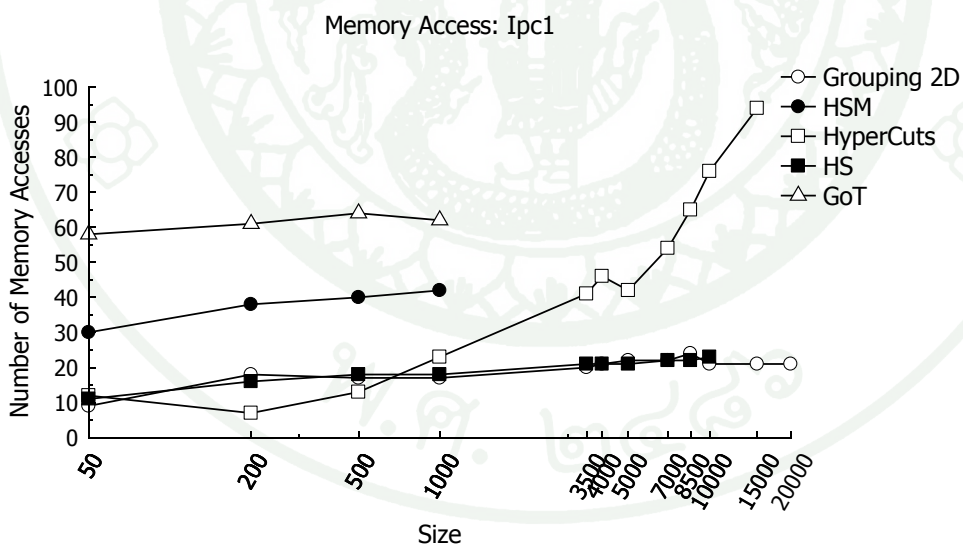
ภาพผนวกที่ ข8 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด FW3 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติ มีจำนวนการเข้าถึงหน่วยความจำสูงสุด 15 ครั้ง



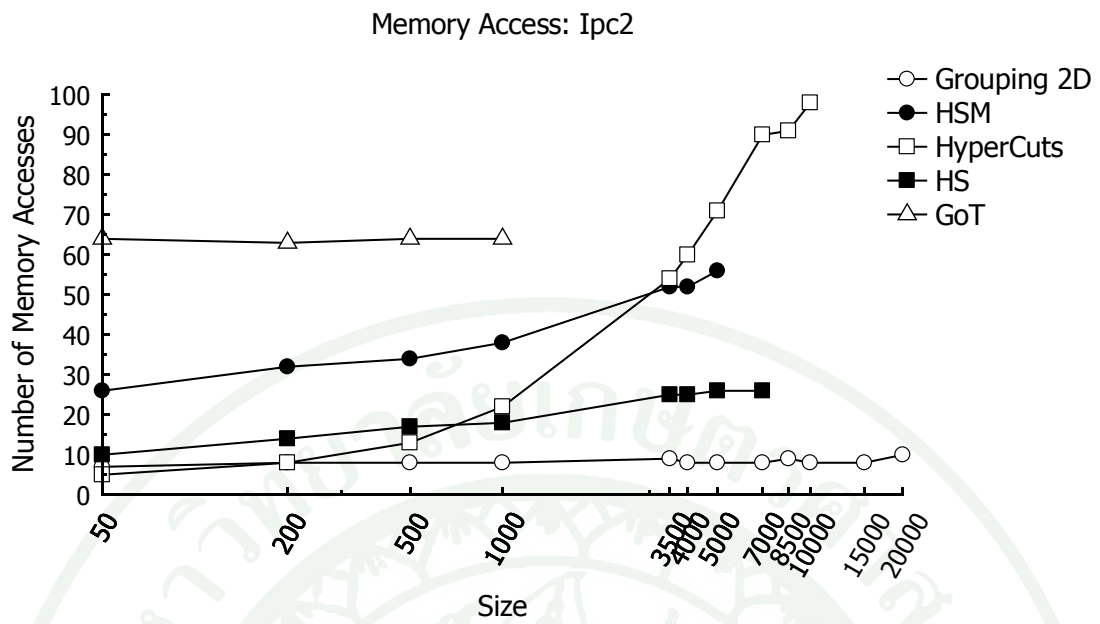
ภาพผนวกที่ ข9 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด FW4 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุด 15 ครั้ง



ภาพผนวกที่ ข10 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด FW5 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุด 20 ครั้ง



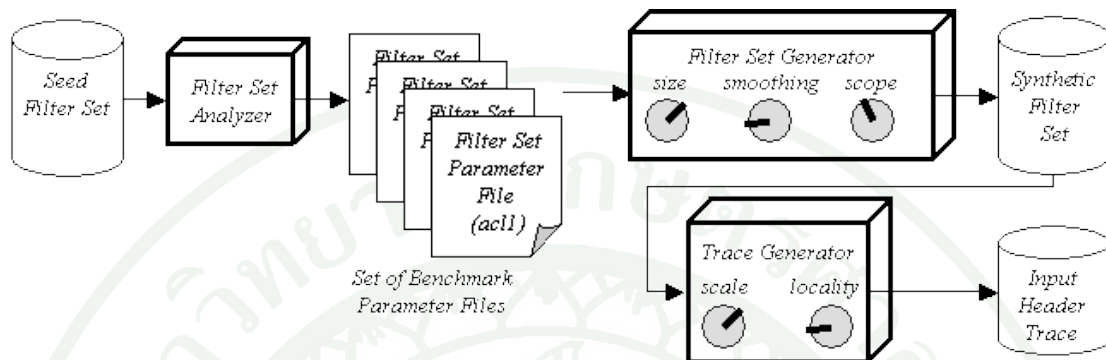
ภาพผนวกที่ ข11 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด IPC1 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติมีจำนวนการเข้าถึงหน่วยความจำสูงสุด 20 ครั้ง



ภาพผนวกที่ ข12 การอ้างอิงหน่วยความจำสำหรับตัวกรองชนิด IPC2 จากภาพ HyperCuts จะมีจำนวนการเข้าถึงหน่วยความจำสูงสุด ในขณะที่การจัดกลุ่มสองมิติจำนวนการเข้าถึงหน่วยความจำสูงสุดแค่ 10 ครั้ง



งานวิจัยนี้ได้สร้างชุดตัวกรองจำลองเพื่อใช้ในการทดสอบขั้นตอนวิธีการคัดแยกแพ็คเกจ โดยผู้วิจัยได้ใช้ ClassBench ซึ่งมีโครงสร้างดังภาพผนวกที่ 25 ซึ่งผู้วิจัยได้ปรับแต่งพารามิเตอร์ เพื่อให้เหมาะสมกับงานวิจัย ดังนี้



ภาพผนวกที่ ค1 โครงสร้าง ClassBench

1. ใช้พารามิเตอร์ไฟล์ 12 ไฟล์ซึ่งเป็นต้นแบบข้อมูลชุดตัวกรองที่ผู้วิจัย ClassBench ได้วิเคราะห์มาจากตัวกรองจริง โดยแบ่งออกเป็นแต่ละชนิดคือ ACL มีพารามิเตอร์ไฟล์ 5 ไฟล์คือ ACL1, ACL2, ACL3, ACL4, และ ACL5, FW มีพารามิเตอร์ไฟล์ 5 ไฟล์คือ FW1, FW2, FW3, FW4, และ FW5 และ IPC มีพารามิเตอร์ไฟล์ 2 ไฟล์คือ IPC1 และ IPC2
2. ปรับแต่งค่าพารามิเตอร์ในการสร้างตัวกรองจำลอง โดยปรับค่า Smooth (ค่าการกระจายตัวของพรีฟิก) เป็นค่า 0 และ 64 และปรับค่า Scope (ค่าความยาวพรีฟิก) โดยปรับไว้ 3 ค่า - 1, 0, 1
3. ปรับค่าพารามิเตอร์ Size (จำนวนข้อมูลตัวกรองในแต่ละชุดตัวกรอง) เพื่อผลการทดสอบงานวิจัยมีความแม่นยำมากที่สุด ผู้วิจัยได้ปรับค่าทั้งหมด 10 ค่าคือ 50, 200, 500, 1k, 3.5k, 8.5k, 10k, 15k, 20k, และ 50k

## ประวัติการศึกษาและการทำงาน

ชื่อ	นายวรา ว่างฉาย
เกิดวันที่	7 สิงหาคม 2529
สถานที่เกิด	อำเภอพุนพิน จังหวัดสุราษฎร์ธานี
ประวัติการศึกษา	วศ.บ. (คอมพิวเตอร์) มหาวิทยาลัยธรรมศาสตร์
ตำแหน่งปัจจุบัน	นักวิเคราะห์ระบบ
สถานที่ทำงานปัจจุบัน	ธนาคารไทยพาณิชย์

