การออกแบบและประเมินสมรรถนะของระบบควบคุมช่องสัญญาณจราจรที่สามารถตรวจจับเพียร์ทูเพียร์เข้ารหัส

Design & Performance Evaluation of a Bandwidth Shaper with Encrypted P2P Detection

ธงชัย เจือจันทร์ และ สมนึก พ่วงพรพิทักษ์

คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม ต.ขามเรียง อ .กันทรวิชัย จ.มหาสารคาม 44150

บทคัดย่อ – เพียร์ทูเพียร์เป็นโพรโทคอลการแชร์ไฟล์ที่ได้รับความนิยมในผู้ใช้อินเทอร์เน็ตในปัจจุบัน ซึ่งการใช้แบนด์วิดท์อย่างตะกละของทราฟฟิกเพียร์ทูเพียร์ ได้สร้างปัญหาต่อระบบควบคุมช่องสัญญาณจราจรในกระบวนการจำแนกและการควบคุม โดยปัจจุบันการเข้ารหัสของทราฟฟิกเพียร์ทูเพียร์มีสัดส่วนสูงขึ้น เพื่อหลีกเลี่ยงการจำแนก ในงานวิจัยก่อนหน้า-นี้ทีมวิจัยได้นำเสนออัลกอริทึมการจำแนกและควบคุมเพียร์ทูเพียร์ที่ถูกเข้ารหัสได้ แม้ว่าอัลกอริทึมดังกล่าวจะเป็นแนวคิดที่ดีกว่างานวิจัยอื่น ๆ แต่ก็ยังมีข้อบกพร่อง คือ การควบคุมผู้ใช้ที่เกินความเหมาะสมและการจำแนกที่ผิดพลาด ในงานวิจัยนี้จึงนำเสนอวิธีแก้ไขข้อบกพร่องดังกล่าว และทำการประเมินบนเครือข่ายทดสอบ ซึ่งแสดงให้เห็นถึงประสิทธิภาพและประสิทธิผลของอัลกอริทึมใหม่จากการออกแบบ

คำสำคัญ – ทราฟฟิกเพียร์ทูเพียร์; โพรโทคอล; สมรรถนะ

ABSTRACT – Peer-to-peer (P2P) file sharing protocols have become popular among the Internet's users. Due to the bandwidth-greediness of the P2P protocols, it is very significant for bandwidth shapers to detect and control them. Yet, P2P traffic encryption has been increasingly used to evade the detection. In our previous study, we have initially proposed an algorithm to detect and shape this encrypted P2P traffic. Although the previous algorithm could outperform the other proposed solution, it still has some drawbacks, namely over-penalty and minor fault negative problems. Hence, we propose in this paper an enhanced algorithm that can solve both weaknesses. The experiment has done on a test-bed to evaluate the enhanced algorithm. The results have demonstrated effectiveness and efficiency of the new algorithm.

KEY WORDS – peer-to-peer traffic; protocol; evaluation

1. บทนำ

ในแต่ละหน่วยงานจะมีช่องสัญญาณทราฟฟิกที่ใช้เชื่อมต่อเข้าสู่อินเทอร์เน็ตจำกัด ระบบควบคุมช่องสัญญาณทราฟิก จึงเป็นอุปกรณ์ที่มีความจำเป็น เพราะการอนุญาตให้โพรโทคอลทุกชนิดใช้อย่างไม่จำกัด จะส่งผลเสียที่รุนแรง เช่น ใช้ช่องสัญญาณทราฟฟิกมากจนส่งผลกระทบต่อโพรโทคอลที่สำคัญได้

ระบบควบคุมช่องสัญญาณทราฟฟิก ต้องใช้กระบวนการจำแนกชนิดเพื่อแยกแยะ และกำหนดขนาดช่องสัญญาณทราฟฟิกให้แตกต่างกันตามลำดับความสำคัญได้ถูกต้อง ซึ่งโดยทั่วไปใช้วิธีจำแนกด้วยหมายเลขพอร์ตและการอ่านเนื้อความของทราฟฟิก

ปัจจุบันการแลกเปลี่ยนไฟล์ด้วยทราฟฟิกประเภทเพียร์-ทูเพียร์ ได้รับความนิยมสูง เพราะสามารถดาวน์โหลดไฟล์ที่ต้องการจากหลายแหล่ง แล้วนำแต่ละส่วนมารวมกัน ดังนั้นการดาวน์โหลดจึงสามารถทำได้รวดเร็ว อีกทั้งการสื่อสารของทราฟฟิกจะสุ่มหมายเลขพอร์ต เพื่อใช้สื่อสาร และในปัจจุบันผู้ใช้สามารถกำหนดรูปแบบการรับส่งข้อมูลให้ถูกเข้ารหัสได้ [1] เช่นโพรโทคอล Bittorrent [2], Gnutella [3] และ Kazaa [4] เป็นต้น ดังนั้นการระบุชนิดโพรโทคอลด้วยหมายเลขพอร์ต และการอ่านเนื้อความทราฟฟิกจึงไม่สัมฤทธิ์ผล ดังงานวิจัยที่ทำการทดสอบเพื่อนำเสนอปัญหาดังกล่าวไว้ก่อนหน้า [5, 6]

ในงานวิจัยนี้ได้ให้ความสำคัญต่อการควบคุมช่องสัญญาณทราฟฟิกเพียร์ทูเพียร์ และผู้ใช้ที่มีพฤติกรรมการใช้ทราฟฟิกที่คล้ายเพียร์ทูเพียร์ เพราะการใช้ลักษณะดังกล่าวจะใช้ช่องสัญญาณทราฟฟิกสูงจนส่งผลเสียต่อองค์กร เมื่อไม่ควบคุมให้มีปริมาณการใช้งานที่จำกัด โดยพัฒนาต่อยอดจากงานวิจัยก่อนหน้านี้ของผู้วิจัย (ISANBC) [7]

งานวิจัย [7] ยังมีปัญหาการจำแนกทราฟฟิกเพียร์ทูเพียร์ที่เข้ารหัสในกรณีการดาวน์โหลด (leeching) เพราะการดาวน์โหลดไฟล์จากปลายทางที่ไม่มีความต่อเนื่อง ทำให้ระบบไม่สามารถตรวจสอบได้ และบทลงโทษผู้ใช้ทราฟฟิกเพียร์ทูเพียร์ที่เกินสมควร (over-penalty) เช่นการหยุดหรือควบคุมการใช้จากหมายเลขไอพีภายในองค์กร ซึ่งเป็นวิธีควบคุมที่ไม่เหมาะสม เพราะผู้ใช้ดังกล่าวอาจใช้โพรโทคอลที่สำคัญอยู่ แต่ถูกระบบควบคุมการใช้ทุกโพรโทคอล จึงเป็นประเด็นปัญหาที่จะแก้ไข ในงานวิจัยนี้จึงเพิ่มความสามารถการจำแนก ทราฟฟิกเพียร์ทูเพียร์กรณีการดาวน์โหลด และการกำหนดบทลงโทษที่ไม่สูงเกินควร เช่น การควบคุมเฉพาะทราฟฟิกเพียร์ทูเพียร์ โดยไม่ส่งผลกระทบต่อการใช้ทราฟฟิกชนิดอื่นของผู้ใช้ในองค์กร

การทดสอบซอฟต์แวร์ ได้ทดสอบเปรียบเทียบกับซอฟต์แวร์เสรี (Open-Source) ที่ใช้วิธีการจำแนกทราฟฟิกด้วยนิพจน์ปรกติอย่าง L7-filter [8] และการจำแนกด้วยหมายเลขพอร์ต จากนั้นทำการประเมินประสิทธิภาพและประสิทธิผลเปรียบเทียบ กับซอฟต์แวร์ต้นแบบจากการออกแบบและพัฒนาต่อยอดจากงานวิจัยก่อนหน้านี้ [7]

ในส่วนที่ 2 จะกล่าวถึงปัญหาและแนวทางแก้ไขปัญหา ส่วนที่ 3 จะกล่าวถึงการออกแบบเพื่อทดลองและประเมินผล ส่วนที่ 4 ส่วนที่ 5 เป็นส่วนอภิปรายผลและบทสรุปตามลำดับ

2. ปัญหาและแนวทางแก้ไขปัญหา

2.1 ปัญหาของทราฟฟิกเพียร์ทูเพียร์

ในทราฟฟิกเพียร์ทูเพียร์มีการแลกเปลี่ยนไฟล์ระหว่างเครื่องลูกข่ายโดยตรง ทำให้ต้องใช้ช่องสัญญาณทราฟฟิกปริมาณสูง การใช้ทราฟฟิกเพียร์ทูเพียร์จึงส่งผลให้โพรโทคอลที่มีความสำคัญ เช่น HTTP ทำหน้าที่ในการเยี่ยมชมเว็บไซต์ทำงานได้ช้าลงหรืออาจใช้การไม่ได้ เนื่องจากทราฟฟิกเพียร์ทูเพียร์ใช้ช่องสัญญาณทราฟฟิกเต็ม

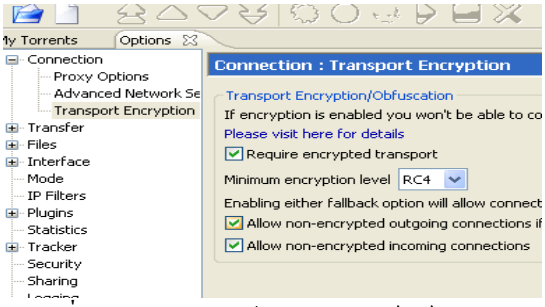
กระบวนการจำแนกทราฟฟิกเพียร์ทูเพียร์ด้วยหมายเลขพอร์ต และการใช้นิพจน์ปรกติไม่สัมฤทธิ์ผล เพราะปัญหาจากการสื่อสารด้วยทราฟฟิกเพียร์ทูเพียร์ จะสุ่มหมายเลขพอร์ตและปิดบังเนื้อความของทราฟฟิกด้วยวิธีเข้ารหัส

มีผู้พัฒนาการจำแนกชนิดทราฟฟิกด้วยพฤติกรรมด้วย Graphlet จากงานวิจัยของ Karagiannis และคณะ [9] แต่วิธีการตรวจสอบข้างต้นยังไม่สามารถนำไปใช้การได้ เพราะการใช้ Graphlet อาจเกิดข้อผิดพลาดที่รุนแรง คือการตรวจจับทราฟฟิกที่ดีเป็นทราฟฟิกเพียร์ทูเพียร์ ดังที่ได้ทำการทดลองและนำเสนอไปแล้วในงานวิจัยก่อนหน้า [7]

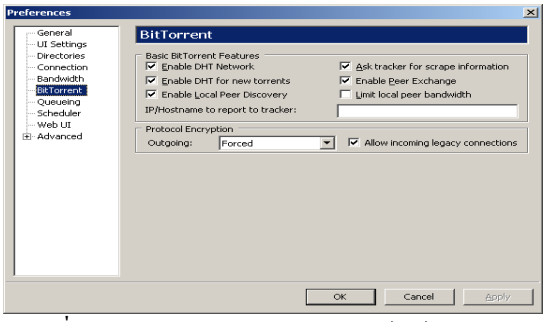
มีผู้เสนอการตรวจสอบโพรโทคอลบิตทอร์เร้นท์ จากงานวิจัยของ Ngiwlay และคณะ [10] ซึ่งเป็นงานวิจัยที่ดีและมีจำนวนความผิดพลาดที่รุนแรง (False Positive) น้อย แต่เวลาที่ใช้ในการตรวจสอบสูง การดาวน์โหลดไฟล์ที่มีขนาดเล็ก แต่ใช้ความเร็วสูงในการดาวน์โหลด ระบบจะไม่สามารถควบคุมบิต

ทอร์เร้นท์ได้ทัน และสามารถตรวจสอบได้เฉพาะโพรโทคอลบิตทอร์เร้นเท่านั้น

ซอฟต์แวร์ที่ใช้งานทราฟฟิกเพียร์ทูเพียร์ ในปัจจุบันหลายซอฟต์แวร์สามารถเข้ารหัสระหว่างการสื่อสารเพื่อหลบเลี่ยงการจำแนก ตัวอย่างเช่น Azureus [11] ดังรูปที่ 1 และ Bittorrent [12] ดังรูปที่ 2 เป็นต้น โดยในซอฟต์แวร์เหล่านี้จะมีตัวเลือกเพื่อปรับแต่งการสื่อสารให้มีการเข้ารหัส



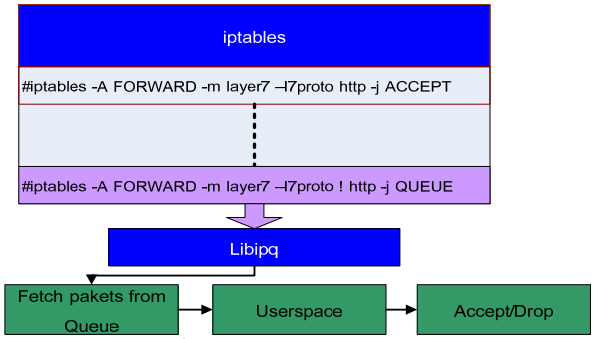
รูปที่ 1. แผงควบคุมการเข้ารหัสของซอฟต์แวร์ Azureus



รูปที่ 2. แผงควบคุมการเข้ารหัสของซอฟต์แวร์ Bittorrent

2.2 การจัดการเครือข่ายด้วย libipq

libipq เป็นไลบรารีที่มีการทำงานร่วมกับ iptables [13] โดยนักพัฒนาสามารถเลือกแพ็คเก็ตที่ต้องการไว้ที่คิว (queue) ของ iptables และพัฒนาซอฟต์แวร์ที่สามารถหยุด (drop) การปรับเปลี่ยน (modify) แพ็คเก็ตได้ โดยตัวอย่างการทำงานระหว่าง iptables และ libipq เมื่อต้องการแพ็คเก็ตที่ไม่ใช่ http ดังรูปที่ 3



รูปที่ 3. การทำงานของ libipq

2.3 การออกแบบ

งานวิจัยก่อนหน้านี้ [7] ผู้วิจัยได้ออกแบบวิธีการจำแนกและระงับการใช้งานทราฟฟิกเพียร์ทูเพียร์ด้วยการใช้วิธีการใช้การตรวจสอบนิพจน์ปรกติ (regular expression) และการเรียนรู้เส้นทางการสื่อสาร

ในงานวิจัยนี้ได้ปรับเปลี่ยนอัลกอริทึมการจำแนกทราฟฟิกเพียร์ทูเพียร์ที่เข้ารหัสในกรณีการดาวน์โหลดไฟล์ และเปลี่ยนวิธีรับแพ็คเก็ตด้วยไลบรารี libpcap เป็น libipq เพราะมีความสามารถหยุด (drop) แพ็คเก็ตที่ต้องการได้ ดังนั้นงานวิจัยนี้จึงสามารถควบคุมทราฟฟิกเพียร์ทูเพียร์ โดยไม่จำกัดการใช้งานระดับหมายเลขไอพีในองค์กร

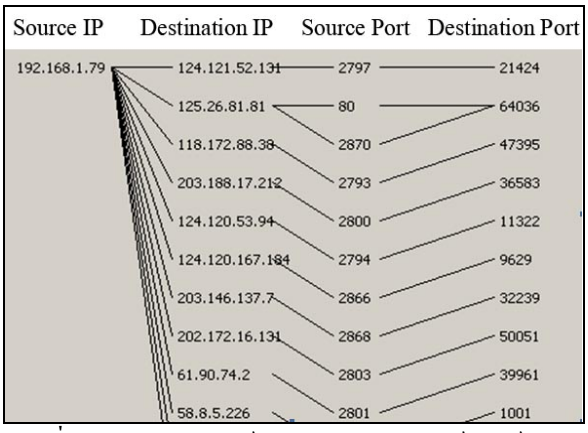
การจำแนกชนิดทราฟฟิกและโพรโทคอลแบ่งเป็น 2 ส่วน คือ (1) การตรวจสอบด้วยการใช้นิพจน์ปรกติและพฤติกรรมตามกลวิธีของงานวิจัยก่อนหน้านี้ (2) การกำหนดนโยบายโดยผู้ดูแลระบบมีดังนี้

1. การจัดการโดยผู้ดูแลระบบ ผู้ดูแลระบบสามารถปรับการทำงานของ ISANBC ได้ดังนี้

1.1 อนุญาตหมายเลขไอพี แม้หมายเลขดังกล่าวถูกใช้งานโดยทราฟฟิกเพียร์ทูเพียร์ แต่เป็นการใช้งานโดยผู้ใช้ที่มีไอพีพิเศษ (VIP IP)

1.2 กำหนดช่วงเวลาในการติดต่อสื่อสารของแต่ละเส้นทาง (t) เพราะในทราฟฟิกเพียร์ทูเพียร์ มีช่วงเวลาในการสื่อสารต่อเนื่องกว่าปกติในกรณีการดาวน์โหลด จึงต้องมีการปรับแต่งเพื่อกำหนดช่วงความต่อเนื่องของการสื่อสารห้ามเกินกว่าที่ผู้ดูแลระบบกำหนด เพราะ Graphlet ของทราฟฟิกเพียร์-ทูเพียร์กรณีนี้จะมีลักษณะคล้ายกับการสื่อสารของผู้ใช้ภายในองค์กรทั่วไป เช่น การใช้โพรโทคอล HTTPS ดังรูปที่ 4 แต่

พฤติกรรมความต่อเนื่องของเวลาในการติดต่อสื่อสารจะแตกต่างกัน



รูปที่ 4. Graphlet การดาวน์โหลดผ่านทราฟฟิกเพียร์ทูเพียร์

1.3 กรณีที่ผู้ใช้ดาวน์โหลดไฟล์ผ่านทราฟฟิกเพียร์ทู-เพียร์จะมีจำนวนครั้งในการติดต่อสื่อสารไปยังปลายทางเป็นจำนวนมาก ดังนั้นจึงต้องกำหนดจำนวนครั้งที่ติดต่อสื่อสาร (c) ของแต่ละเส้นทาง ซึ่งเป็นพารามิเตอร์เพิ่มเติมงานวิจัยก่อนหน้า [7] เพื่อใช้จำแนกทราฟฟิกเพียร์ทู-เพียร์กรณีดาวน์โหลด

1.4 ผู้ใช้ที่มีจำนวนเซสชัน (session) สูง และสื่อสารแบบเข้ารหัสระบบจะตรวจสอบจำนวนเซสชัน (k) เพราะทราฟฟิกเพียร์ทูเพียร์จะสร้างจำนวนเซสชันต่อผู้ใช้สูง

2. วิธีจำแนกชนิดทราฟฟิก

การจำแนกชนิดทราฟฟิกทำได้ดังรูปที่ 5 ซึ่งมีกระบวนการดังนี้

2.1 จำแนกไอพีผู้ใช้ระดับพิเศษ จากการกำหนดโดยผู้ดูแลระบบ

2.2 การจำแนกด้วยนิพจน์ปรกติด้วยซอฟต์แวร์ L7-filter ในโพรโทคอลที่ไม่ถูกเข้ารหัส ส่วนโพรโทคอลที่มีการเข้ารหัส จะกำหนดเป็นแพ็คเก็ตต้องสงสัยและเก็บไว้ที่คิวของ iptables เพื่อใช้ตรวจสอบพฤติกรรมในข้อ 2.3

2.3 การจำแนกด้วยพฤติกรรม เป็นวิธีการเรียนรู้เส้นทางการสื่อสาร โดยต้องเก็บเส้นทางการสื่อสารคือหมายเลขพอร์ตต้นทาง-ปลายทาง หมายเลขไอพีต้นทาง-ปลายทาง (flow) และผู้ดูแลระบบต้องกำหนดหน่วยเวลา (Traffic Collecting Delay: TCD) ที่ใช้จัดเก็บข้อมูลเป็นเวลานานเท่าใด ภายหลังเก็บข้อมูลจึงเริ่มทำการเรียนรู้เส้นทางและจำแนกชนิดทราฟฟิก (Learning Delay: LD)

2.4 การจำแนกด้วยพฤติกรรมแบ่งเป็นสองแบบคือ

- กรณีการอัพโหลดไฟล์ (seeding) กลไกการทำงานสามารถศึกษาได้จากงานวิจัยก่อนหน้านี้ของผู้วิจัย [7]

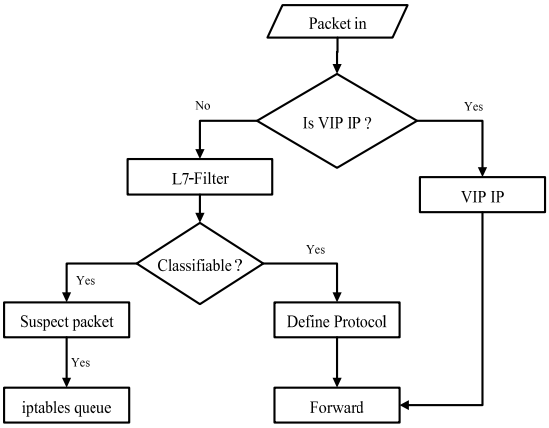
- กรณีดาวน์โหลดไฟล์ จะใช้ช่วงเวลาการสื่อสาร ซึ่งมีพฤติกรรมที่โดดเด่นดังกล่าวไว้ในข้อ 1.2 ซึ่งได้เพิ่มการนับจำนวนครั้งในการติดต่อสื่อสาร เพราะการตรวจสอบการสื่อสารที่ต่อเนื่อง ยังมีปัญหาการตรวจสอบ (trace) ข้อมูล ดังนั้นการตรวจสอบจากการนับจำนวนครั้งที่สื่อสาร จะช่วยให้การจำแนกทราฟฟิกเพียร์ทูเพียร์ กรณีการดาวน์โหลดให้มีความแม่นยำยิ่งขึ้น ซึ่งมีกลไกการทำงานดังรูปที่ 6 โดยกำหนด n คือจำนวนเส้นทางการสื่อสารที่เข้ารหัส และกระบวนการทำงานมีดังนี้

- รับแพ็คเก็ตจากคิวของ iptables และทำการสร้าง Graphlet
- นับจำนวนครั้งการสื่อสารของแต่ละเส้นทางการสื่อสาร (flow) ซึ่งเป็นค่าที่เริ่มนับใหม่เมื่อครบรอบ TCD
- ตรวจสอบจำนวนเซสชันของผู้ใช้ว่ามีการติดต่อไปยังปลายทางมากกว่ากำหนด (k) หรือไม่ (1) ถ้ามากกว่าจะทำการตรวจสอบเพิ่มเติม (2) ถ้าน้อยกว่าจะส่งต่อ เพราะไม่เป็นพฤติกรรมของไฟล์แชร์ริ่ง และไม่รบกวนผู้ใช้อื่นในองค์กร
- จำแนกทราฟฟิกเพียร์ทูเพียร์ จากความต่อเนื่องของแต่ละเส้นทางการสื่อสารที่มากกว่า t
  - o เส้นทางที่มีค่ามากกว่า t จะจำแนกเป็นทราฟฟิกเพียร์ทูเพียร์
  - o ถ้าน้อยกว่าจะถูกจำแนกในขั้นตอนถัดไป
- ตรวจสอบทราฟฟิกเพียร์ทูเพียร์จากการนับจำนวนครั้งในการสื่อสาร จะตรวจสอบว่ามีจำนวนครั้งมากกว่า c หรือไม่
  - o ถ้ามากกว่าจะจำแนกเป็นทราฟฟิกเพียร์ทูเพียร์
  - o ถ้าน้อยกว่าจะระบุว่าไม่เป็นทราฟฟิกเพียร์ทูเพียร์และส่งแพ็คเก็ตต่อไปยังปลายทาง

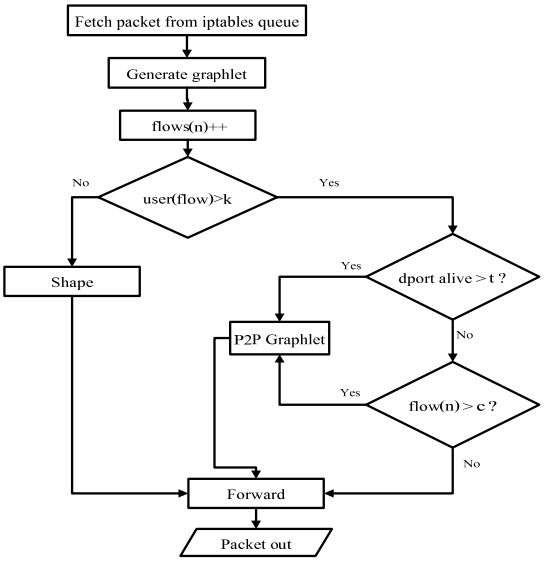
หลังการจำแนกระบบจะทราบเส้นทางการสื่อสารของทราฟฟิกเพียร์ทูเพียร์ จะทำการควบคุมดังนี้

1. ควบคุมให้ใช้งานตามช่องสัญญาณจราจรที่ผู้ดูแลระบบกำหนด

2. ตรึอบเส้นทางการสื่อสารทั้งหมด เพื่อลดการรบกวนการใช้ ทราฟฟิกอื่น



รูปที่ 5. การทำงานโดยรวมของระบบ



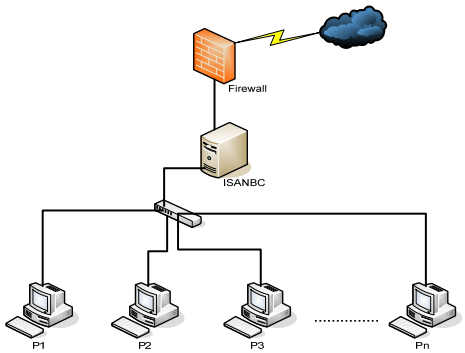
รูปที่ 6. จำแนกกรณี leeching ใหม่

3. การออกแบบเพื่อทดลองและประเมินผล

3.1 ออกแบบการทดลอง

การทดลองได้ออกแบบแผนผังการเชื่อมต่อระบบเครือข่ายดาวน์โหลดและอัพโหลดโพรโทคอลที่แตกต่างกัน โดยใช้เครื่องคอมพิวเตอร์ลูกข่ายจำนวน 6 เครื่อง แบ่งกลุ่มโพรโทคอลออกเป็น 3 ประเภทดังตารางที่ 1 และสร้างแบบจำลองเครือข่ายดังรูปที่ 7 เครื่องคอมพิวเตอร์ที่ทำหน้าที่ควบคุมทราฟฟิกคือเกตเวย์ที่มีสมรรถนะ ซีพียู (CPU) PENTIUM 4.0 GHZ หน่วยความจำหลักขนาด 2 GB

การใช้งานทราฟฟิกในการทดลอง จะใช้ตามธรรมชาติของทราฟฟิกในปัจจุบัน เช่น ทราฟฟิกเพียร์ทูเพียร์ จะมีการสุ่มหมายเลขพอร์ต และเว็บจะใช้พอร์ตหมายเลข 80 ในการสื่อสาร เป็นต้น



รูปที่ 7. แผนผังการเชื่อมต่อระบบ

การแบ่งกลุ่มทราฟฟิกเป็น 3 ประเภท เพราะผู้วิจัยได้วิเคราะห์ลักษณะการสื่อสารออกเป็น (1) สตรีมมิ่ง เป็นลักษณะการสื่อสารที่มีความต่อเนื่องยาวนานและคล้ายกับทราฟฟิกเพียร์ทูเพียร์แต่จะไม่มีการติดต่อสื่อสารกัน ระหว่างลูกข่าย (2) เพียร์ทูเพียร์เป็นทราฟฟิกที่ติดต่อสารกันแบบลูกข่ายต่อลูกข่าย ดังที่อธิบายแล้วข้างต้น (3) ทราฟฟิกอื่น ๆ ดังตารางที่ 1 เพื่อใช้ทดสอบสมรรถนะการจำแนกและความสามารถในการควบคุมทราฟฟิกของซอฟต์แวร์ทดสอบ

เครื่องคอมพิวเตอร์ที่นำมาทดสอบเป็นคอมพิวเตอร์ส่วนบุคคล (PERSONAL COMPUTER: PC) ใช้งานโพรโทคอลที่แตกต่างกันพร้อมกันดังตารางที่ 2 การตรวจสอบสมรรถนะการจำแนกโดยผลลัพที่คาดว่าจะได้รับเบื้องต้นคือ PC1, PC2 และ PC3 เท่านั้นที่ใช้งาน ทราฟฟิกเพียร์ทูเพียร์ และสามารถหยุดหรือควบคุมทราฟฟิกเพียร์ทูเพียร์ได้

การทดสอบ แบ่งการจำแนกเป็น 3 วิธีคือจำแนกด้วยนิพจน์ปรกติ หมายเลขพอร์ตมาตรฐานและ ISANBC

ตารางที่ 1. กลุ่มทราฟฟิกและโพรโทคอล

	ชนิดทราฟฟิก		
	Streaming	P2P	Others
Protocol	Msnp, TCP	Gnutella, eDonkey, Bittorrent, WPNP	HTTP, FTP SSH
Application	GStreamer, Skype, Windows Live	Shareza, eMule, WinMX, Bittorrent	Firefox, IE, Filezilla, Secure Shell Client

ตารางที่ 2. กำหนดการใช้งานโพรโทคอลแต่ละ PC

Environment			
	Network type		
	Streaming	P2P	Others
PC 1	✗	✓	✓
PC 2	✗	✓	✗
PC 3	✓	✓	✗
PC 4	✓	✗	✗
PC 5	✓	✗	✓
PC 6	✗	✗	✓

3.2 เกณฑ์ที่ใช้ประเมินผล

ตารางที่ 3. การจำแนก

ผลการจำแนก / ทราฟฟิก	จำแนกเป็น P2P	จำแนกไม่เป็น P2P
P2P	TRUE POSITIVE (TP)	FALSE NEGATIVE (FN)
ไม่ใช่ P2P	FALSE POSITIVE (FP)	TRUE NEGATIVE (TN)

การประเมินผลสมรรถนะของซอฟต์แวร์ทำโดยการทดลองจำนวน 30 ครั้ง ให้ใช้งานโพรโทคอลต่างๆ ไม่ซ้ำกันตามทราฟฟิกแต่ละกลุ่มเป็นเวลา 10 นาที และประเมินความถูกต้อง และเวลาที่ใช้ในการจำแนกของ ISANBC ซึ่งเกณฑ์การประเมินความถูกต้องสามารถตรวจสอบได้ตามตารางที่ 3 ดังนี้

1. TP แสดงให้เห็นถึงการจำแนก ทราฟฟิกเพียร์ทูเพียร์ที่เข้ารหัสได้ถูกต้อง

2. FN เป็นการจำแนกทราฟฟิก เพียร์ทูเพียร์ที่เข้ารหัสไม่ได้ โดย ทราฟฟิกจริงเป็นเพียร์ทูเพียร์แต่การจำแนกพบเป็นทราฟฟิกทั่วไป

3. FP เป็นการจำแนกทราฟฟิกทั่วไปเป็นเพียร์ทูเพียร์ที่เข้ารัหส FP เป็นข้อผิดพลาดที่รุนแรงที่สุดเนื่องจากทำให้ ทราฟฟิกที่ดีถูกระบบควบคุมการใช้งานเหมือนกับทราฟฟิกเพียร์ทูเพียร์ได้

4. TN เป็นการจำแนกทราฟฟิกทั่วไปได้ถูกต้อง

ตารางที่ 4. ความสามารถการควบคุมทราฟฟิก *ISANBC เปรียบเทียบกับวิธีอื่น*

	Encrypted P2P	P2P
ISANBC	✓	✓
L7-filter	✗	✓
Port base	✗	✗

จากตารางที่ 4 แสดงถึงประสิทธิภาพในการจำแนกทราฟฟิกเพียร์ทู-เพียร์ที่เข้ารหัสและไม่เข้ารหัส ซึ่ง ISANBC ใหม่สามารถจำแนกได้ทั้งสองรูปแบบ และหลังการจำแนกจะสามารถเลือกควบคุมได้ 2 วิธีคือ

1. การหยุดการใช้ทราฟฟิกเพียร์ทูเพียร์ จากการทดสอบพบว่าหลังจากตรวจสอบพบสามารถหยุดการสื่อสารเพียร์ทูเพียร์ได้ทันที และการใช้งานทราฟฟิกอื่น ๆ ยังเป็นปกติ

2. การจำกัดการใช้ทราฟฟิกเพียร์ทูเพียร์ ISANBC สามารถควบคุมการใช้ทราฟฟิกเพียร์ทูเพียร์ได้ตามที่ผู้ดูแลระบบกำหนดได้

4. อภิปรายผล

ประสิทธิภาพของอุปกรณ์ที่ใช้ควบคุมทราฟฟิก ที่ผู้ดูและระบบต้องการนำไปใช้กับเครือข่ายขนาดใหญ่เช่น เครือข่ายของมหาวิทยาลัย จำเป็นต้องใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพการทำงานสูง เพื่อไม่ให้ส่งผลกระทบกับกับค่า LD

การพัฒนาด้วยไลบรารี libipq จะเกิดปัญหากับขนาดคิวของ iptables ซึ่งมีค่าดีฟอลท์เท่ากับ 1024 แต่เมื่อ Bandwidth Shaper ถูกนำไปใช้กับเครือข่ายขนาดใหญ่ ควรทำการขยายขนาดของคิวให้เหมาะกับขนาดเครือข่ายในองค์กร จึงไม่เกิดปัญหาคิวของ iptables เต็ม

5. บทสรุป

จากงานวิจัยนี้เป็นงานวิจัยที่พัฒนาระบบต่อยอดจากงานวิจัยก่อนหน้าและดีขึ้นดังนี้

1. เพิ่มความสามารถในการตรวจสอบทราฟฟิกเพียร์ทูเพียร์กรณีการดาวน์โหลดไฟล์ และทำให้การจำแนกทราฟฟิกเพียร์ทูเพียร์มีความแม่นยำสูงขึ้น

2. เพิ่มความสามารถในการควบคุมทราฟฟิกเพียร์ทูเพียร์ โดยทำให้การจำกัดผู้ใช้ภายในองค์กรไม่มากเกินสมควร (OVER-PENALTY) และยังควบคุมการสื่อสารที่ต่อเนื่องของโพรโทคอลที่เข้ารหัสเช่น SFTP ซึ่งเป็นไฟล์แชร์-ริ่งที่ผู้ใช้สามารถสื่อสารแบบเซสชัน (MULTI-SESSION) ได้

ระบบมีประสิทธิภาพ โดยในการทดลองแต่ละครั้งใช้เวลา 10 นาที ระบบจะสามารถตรวจสอบพบใช้เวลาเฉลี่ยประมาณ 2 นาที จากการกำหนดให้ตรวจสอบทราฟฟิกทุก ๆ 2 นาที แสดงถึงประสิทธิภาพของเวลาในการจำแนกเร็วกว่างานวิจัยก่อนหน้า และเป็นค่าที่เหมาะสมจากการทดสอบของผู้วิจัย โดยผู้ดูแลระบบสามารถปรับแต่งให้เหมาะสมกับ ทราฟฟิกของแต่ละองค์กรได้ ส่วนเวลาที่ใช้ในการวิเคราะห์พฤติกรรมจะใช้เวลาน้อยกว่า 10 วินาที

ประสิทธิผลจากงานวิจัย สามารถตรวจสอบทราฟฟิกเพียร์ทูเพียร์ได้แม้มีการเข้ารหัส และจากการประเมินพบ FN แต่เป็น FN ที่เกิดจากเพียร์ทูเพียร์ที่พยายามติดต่อกับลูกข่ายอื่น ๆ และเพียร์ทูเพียร์ที่ดาวน์โหลดไฟล์ขนาดเล็ก ซึ่งเป็นส่วนผิดพลาดที่ไม่ส่งผลกระทบต่อองค์กร

เอกสารอ้างอิง

[1] BitTorrent, "BitTorrent protocol encryption." Vol. 2009, 2009.

[2] D. Harrison and S. Shalunov, "BitTorrent Local Tracker Discovery Protocol," www.bittorrent.org, 2008.

[3] P. Kirk, "Gnutella Protocol," rfc-gnutella.sourceforge.net, 2003.

[4] Kazaa, http://www.kazaa.com, 2009.

[5] H. Kim, K. Claffy, M. Fomenkov, N. Brownlee, D. Barman, and M. Faloutsos, "Comparison of Internet Traffic Classification Tools", *IMRG WACI*, 2007.

[6] A. Madhukar and C. Williamson, "A Longitudinal Study of P2P Traffic Classification," University of Calgary, 2005.

[7] T. Chuachan and S. Puangpronpitag, "P2P Traffic Shaping by Checking Regular Expression and Learning Traffic Flow," *National Computer Science and Engineering Conference* Bangkok, Thailand, November 2009.

[8] l7-filter, "http://l7-filter.sourceforge.net," 2009.

[9] T. Karagiannis, K. Papagiannaki, and M. faloutsos, "Multilevel Traffic Classification in the dark," *ACM SIGCOMM*, Pennsylvania, 2005.

[10] N. Wanchai, I. Chalermek, and T.-a. Yunyong, "Bittorrent peer identification based on behaviors of a choke algorithm," *Proceedings of the 4th Asian Conference on Internet Engineering* Pratunam, Bangkok, Thailand: ACM, 2008.

[11] Azureus, http://azureus.sourceforge.net, 2007.

[12] Bittorrent, http://www.bittorrent.com/, 2009.

[13] netfilter, http://www.netfilter.org, 2009.